User's Manual

Climate Forecast System Version 2.0

Center for Development of Advanced Computing
August, 2014

Version: 1.0	Date: Aug. 15, 2014

User's Manual

Climate Forecast System Vn. 2.0

CDAC Knowledge Park,
No. 1, Old Madras Road, Byappanahalli,
Bengaluru – 560038, Karnataka (India)
Ph.: +91-80-6611 6400/ 01/ 02/ 03,
25244059,25246823,25246826
Fax: +91-80-25247724
www.cdacb.in

Revision History

Version No.	Date	Revision Description	Author
1.0	15/08/2014	User's Manual for Climate Forecast System Vn. 2.0	Mohit Ved, Ramesh Naidu

Contents

1. General Information

1.1 Preamble

This User's Manual describes the Climate Forecast System Version 2.0 (CFSv2), a fully coupled ocean-land-atmosphere dynamical seasonal prediction system that became operational at NCEP in March, 2011. CFSv2 is the successor of the earlier version of the model named CFSv1, which became operational in August, 2004. This version has upgrades to nearly all aspects of the data assimilation and forecast model components of the system. The model uses the shell environment for operation and does not come with any GUI. The model has been programmed using FORTRAN 77 with batch input/output routines written in C language. The code also makes use of Message Passing Interface constructs. The model is used for long range and seasonal forecasting.

1.2 Organization of the manual

Section 1: General information with regards to the manual, abbreviations used in the manual and the point of contact for using the manual and the related product.

Section 2: Brief specifications of the model physics including components of the model, upgrades to the previous version and the design of Retrospective forecasts carried out and the operational configuration at NCEP.

Section 3: A brief description of the directories included in the package, a description of the functionalities of the modules and scripts, along with the code flow of the forecast cycle of the model and the post-processing.

Section 4: General information about the Linux cluster at CDAC, installation procedure and guidelines for running the model in coupled mode, and a description of the outputs generated by the model and the post-processor program. This is one of the reference platform on which the model is tested successfully.

1.3 Acronyms and Abbreviations

AMIP	: Atmospheric Model Inter-comparison Project
CDAC	: Centre for Development of Advanced Computing
CDAS	: Climate Data Assimilation System
CFS	: Climate Forecast System
CPC	: Climate Prediction Center
GARUDA	: India's first national grid initiative
GFDL	: Geophysical Fluid Dynamics Laboratory
GFS	: Global Forecast Model
GODAS	: Global Ocean Data Assimilation System
GPCP	: Global Precipitation Climatology Project
ISMR	: All India Summer Monsoon Rainfall
MOM	: Modular Ocean Model
MRF	: Medium Range Forecast
NCAR	: National Center for Atmospheric Research
NCEP	: National Centers for Environmental Prediction
NOAA	: National Oceanic and Atmospheric Administration
SFM	: Seasonal Forecast Model
SST	: Sea Surface Temperature

1.4 Point of Contact

All queries related to the CFS model can be sent to spim@cdac.in, or can also be mailed at the following address:

c/o Seasonal Prediction of Indian Monsoon (SPIM) Team,
C-DAC Knowledge Park,
No. 1, Old Madras Road,
Byappanahalli,
Bengaluru – 560038,
Karnataka (India)

2. Describing the System

2.1 Introduction to CFSv2

The first release of CFS, retroactively called CFSv1, was implemented into operations at NCEP in August 2004 and was the first quasi-global, fully coupled atmosphere-ocean-land model used at NCEP for seasonal prediction (Saha et al., 2006). CFSv1 was developed from four independently designed pieces of technology, namely the R2 NCEP/DOE Global Reanalysis (Kanamitsu et al., 2002) which provided the atmospheric and land surface initial conditions, a global ocean data assimilation system (GODAS) operational at NCEP in 2003 (Behringer, 2007) which provided the ocean initial states, NCEP's Global Forecast System (GFS) operational in 2003 which was the atmospheric model run at a lower resolution of T62L64, and the MOM3 ocean forecast model from GFDL.

CFSv2 has improvements in all four components mentioned above, namely the two forecast models and the two data assimilation systems. CFSv2 also has a few novelties: an upgraded four level soil model, an interactive three layer sea-ice model, and prescribed historical (i.e. rising) CO_2 concentrations. But above all, CFSv2 was designed to improve consistency between the model states and the initial states produced by the data assimilation system.

The atmospheric component of CFSv2 is the NCEP atmospheric GFS model, (Moorthi et al. 2001), with significant improvements. GFS is a global spectral model. A sample of improvements is listed here. For complete information on improvements refer to Saha et al. 2012. The atmospheric model has a spectral triangular truncation of 126 waves (T126) in the horizontal (equivalent to nearly a 100 Km grid resolution) and a finite differencing in the vertical with 64 sigma-pressure hybrid layers. The vertical coordinate is the same as that in the operational CDAS. Additional improvements to the GFS Atmospheric Model (AM) include fast and accurate LW radiation parameterization based on the RRTM developed at AER (Mlawer et al. 1997). It is also coupled to a four-layer NOAH Land Surface Model (Ek et al. 2003) and a two-layer Sea Ice Model (Wu et al. 2005). In addition to gravity wave drag, the model now includes a parameterization of mountain blocking (Alpert 2004) following the subgrid-scale orographic drag parameterization by Lott and Miller (1997). The GFS AM now takes advantage of the ESMF-based modern computer algorithms (Collins et al. 2005). An update of the ozone production and destruction terms is done using the monthly mean data provided by the NRL (McCormack et al. 2006).

The ocean and sea-ice models are identical to those used in CFSR (Saha et al., 2010). The oceanic component is the MOM version 4p0d (), which is a finite difference version of the ocean primitive equations configured under the Boussinesq and hydrostatic approximations. The model uses the tri-polar grid developed by Murray (1996). Northward of 65°N it uses a rotated bipolar grid that places two poles over land, thus eliminating the singularity in the northern ocean, while southward of 65°N it uses a regular latitude × longitude grid. The horizontal layout is a staggered Arakawa B grid and geometric height is in the vertical. The ocean surface boundary is computed as an explicit free surface. The zonal resolution is 1/2°. The meridional resolution is 1⁄4° between 10°S and 10°N, gradually increasing to 1/2° poleward of 30°S and 30°N. There are 40 layers in the vertical, with 27 layers in the upper 400 m, and the bottom depth is approximately 4.5 km. The vertical resolution is 10 m from the surface to the 240-m depth, gradually increasing to about 511 m in the bottom layer. Vertical mixing follows the nonlocal K-profile parameterization of Large et al. (1994). The horizontal mixing of tracers uses the iso-neutral method developed by Gent and McWilliams (1990; Griffies et al. 1998). The horizontal mixing of momentum uses the nonlinear scheme of Smagorinsky (Griffies and Halberg 2000). The configuration for the MOM4p0d is similar to the one used for MOM version 3 with CFSv1 (Saha et al. 2006), but the resolution has been approximately doubled and the MOM4p0d is fully global with an Arctic Ocean and an interactive ice model, whereas MOM3 is truncated at 64°N and 74°S.

The sea ice model is from the GFDL Sea Ice Simulator, with some modifications. Its model grid is identical to the ocean model grid; there are three layers for the sea ice model, including two equal layers of sea ice and one layer of snow. In each ice grid there are five categories of possible sea ice thicknesses (0 – 0.1, 0.1 – 0.3, 0.3 – 0.7, 0.7 – 1.1 m, and the category greater than 1.1 m). Sea ice dynamics are based on Hunke and Dukowicz (1997) using the elastic–viscous–plastic technique to calculate ice internal stress. Ice thermodynamics are based on Winton (2000).

The LSM used in CFSv2 is the NOAH Land Surface Model (LSM) (Ek et al., 2003), which was first implemented in the GFS for operational medium-range weather forecasts (Mitchell et al., 2005) and then in the CFSR (Saha et al., 2010). Within CFSv2, NOAH LSM is employed in both the coupled land-atmosphere-ocean model to provide land-surface prediction of surface fluxes (surface boundary conditions), and in the Global Land Data Assimilation System (GLDAS) to provide the land surface analysis and evolving land states.

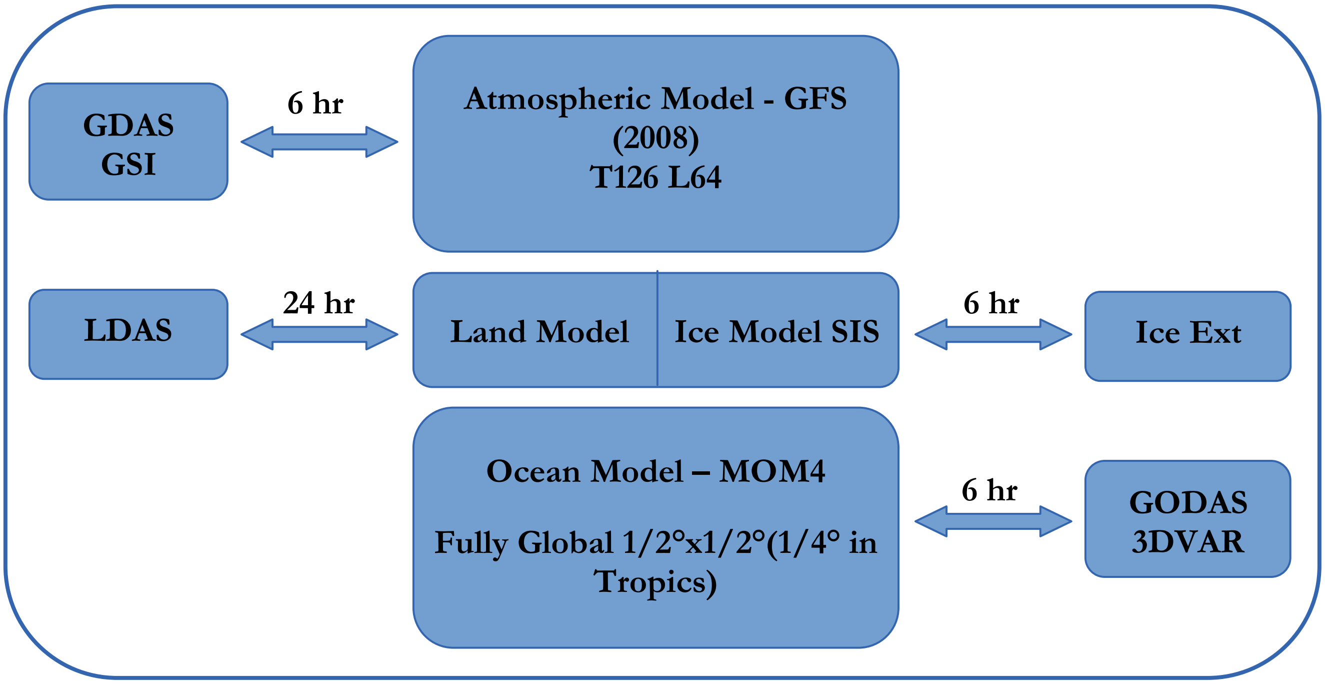


Figure 2.1. The main components of the Climate Forecast System Vn. 2.

The CFS model, which runs on multiple processes with message-passing tools, uses a parallel-programming model called MPMD. The three programs in the CFS: the atmospheric model (GFS), the ocean model (MOM4), and the coupler, each of which has its own data flow, run independently, but they exchange data as follows: the GFS runs on the atmospheric time step Δa (3 min), MOM4 runs on a fast sea ice time step Δi (also 3 min) for the sea ice model, and a slow ocean time step Δo (30 min) for both the ocean model and slow sea ice time step, while the coupler runs on a time step Δc, where Δc = max (Δa, Δi). At every coupler time step Δc, the coupler will receive data from both the GFS and MOM4 sea ice model and will send the needed data back to them, respectively. At every ocean time step, in addition to the data exchanged between GFS and sea ice, the coupler will also receive accumulated variables (fluxes) from the GFS and send them to the ocean model, while receiving data from the ocean and sea ice model and sending them back to the GFS.

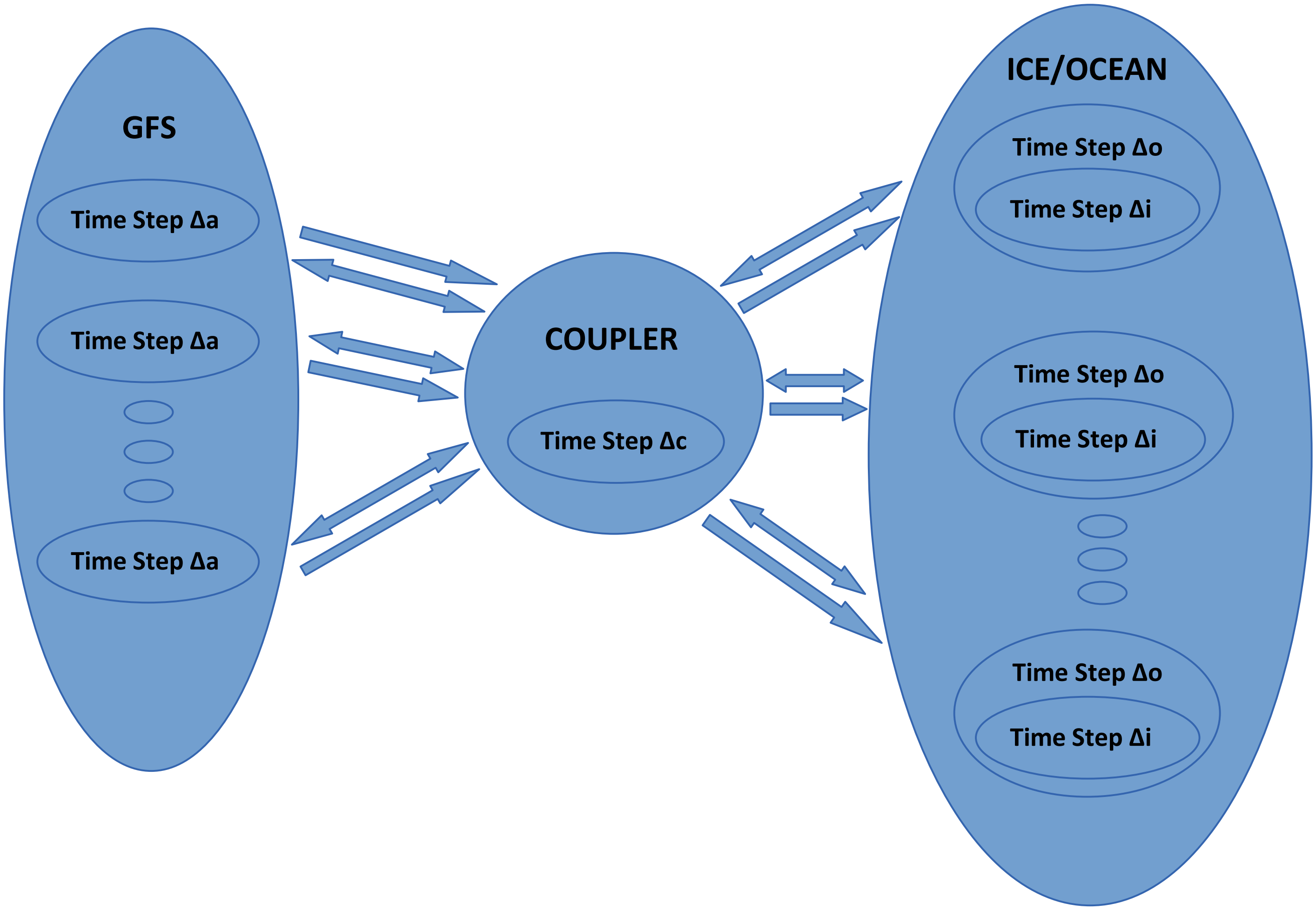


Figure 2.2: GFS-SeaIce/MOM4 Coupler

The CFSv2 model in a nutshell is described in the following table:

Attribute	**v2 Configuration**
Analysis Resolution	38 km / 64 levels
Atmosphere model	2008/100 km/64 levels Variable CO_2 AER SW & LW radiation Prognostic clouds & cloud condensate Re-tuned mountain blocking Convective gravity wave drag
Ocean model	MOM-4 fully global ¼ x ½ deg. Assimilated depth 4737 m
Land surface model (LSM) and assimilation	4 level NOAH model GLDAS driven by observed precipitation
Sea ice	Daily analysis and Prognostic sea ice
Coupling	30 minutes
Data assimilation	Radiances assimilated, 2008 GSI, coupled background

Table 2.1 CFSv2 configuration chart

2.2 Upgrades to CFSv1

There have been significant modifications in CFSv1 to build CFSv2. It took seven years for the developers to implement the following aspects in CFSv2.

- Introducing a 3-layer interactive global sea ice model, as well as a global land data assimilation.
- Increase in the resolution of the atmospheric forecast model from T62 (210 km) to T126 (100 km).

- Upgrading the ocean forecast model from the limited area GFDL MOM3 to the global MOM4. The horizontal resolution is increased from 0.33° to 0.25° from 10° North to 10° South latitudes. Northwards and southwards to the poles, the resolution is increased from 1° to 0.5° globally.
- Upgrading the land surface model from 2-level OSU model to 4-level NOAH land model.
- Upgrading the data assimilation for the climate forecast model. The resolution of the atmospheric Climate Data Assimilation System version 2 (CDAS2) is upgraded from T62 (210 km) with 28 sigma levels to T574 (27km) with 64 hybrid sigma-pressure levels.
- Changing the Spectral Statistical Interpolation scheme (SSI) to a Gridpoint Statistical Interpolation scheme (GSI).
- Assimilating Satellite radiances directly, instead of retrievals.
- Upgrading the Global Ocean Data Assimilation (GODAS) from MOM3 to MOM4.
- In addition, introducing a new Global Land Data Assimilation (GLDAS) which uses observed CPC precipitation as forcing for the NOAH land model.
- Significant additions to parameters in the pressure GRIB (pgb), flux files (flx) and ocean (ocn) files.
- In addition, there is a new file that contains parameters on Isentropic surfaces (ipv).
- Significant changes to the format and content of all the model files, due to the increases in resolution.
    - The horizontal resolution of *pgb* files is increased from 2.5° x 2.5° to 1° x 1° and the number of pressure levels is increased from 17 to 37.
    - The size of the flux file is increased from the Gaussian grid for T62 (192 x 94) to that for T126 (384 x 190).
    - The ocean file is increased from 2.5° x 2.5° to 0.5° x 0.5°.
    - The new Isentropic file has a resolution of 1° x 1°.
- There is also an increase in the temporal resolution of the output forecast data, from 12-hourly to 6-hourly.

2.3 CFSv2 Retrospective Forecasts at NCEP

Alike CFSv1, the CFSv2 also includes a comprehensive set of retrospective runs that are used to calibrate and evaluate the skill of its forecasts. These runs are available as 9-months, first season and 45-days retrospective forecasts.

2.3.1 9-month Retrospective Predictions

The retrospective 9-month forecasts have initial conditions of the 0, 6, 12 and 18Z cycles for every 5th day, starting from 1 Jan 0Z of every year, over the 29-year period i.e. 1982-2010. This is required to calibrate the operational CPC longer-term seasonal predictions (ENSO, etc.). There are 292 forecasts for every year for a total of 8468 forecasts. This results in an ensemble size of 24 forecasts for each month, except November which has 28 forecasts.

The retrospective forecast calendar (Saha et. al. 2013) outlines the forecasts that are used each calendar month, to estimate proper calibration and skill estimates, in such a way to mimic CPC operations. Smoothed calibration climatologies have been prepared from the forecast monthly means and time series of selected variables and is available for download together with selected data from the retrospective forecasts from the NCDC web servers (Saha et. al. 2013)

2.3.2 First Season and 45-day Retrospective Forecasts

These retrospective forecasts have initial conditions from every cycle (0, 6, 12 and 18Z) of every day over the 12-year period from Jan 1999-Dec 2010. Thus, there are approximately 365*4 forecasts per year, for a total of 17520 forecasts. The forecast from the 0Z cycle was run out to a full season, required to calibrate the operational CPC first season predictions for hydrological forecasts (precip, evaporation, runoff, streamflow, etc), while the forecasts from the other 3 cycles (6, 12 and 18Z) were run out to exactly 45 days, required for the operational CPC week3-week6 predictions of tropical circulations (MJO, PNA, etc)(Saha et. al. 2013). Smoothed calibration climatologies have been prepared from the forecast time series of selected variables (http://cfs.ncep.noaa.gov/cfsv2.info/CFSv2.Calibration.Data.doc) and is available for download. It is essential that some smoothing is done when preparing the climatologies of the daily time-series, which are quite noisy.

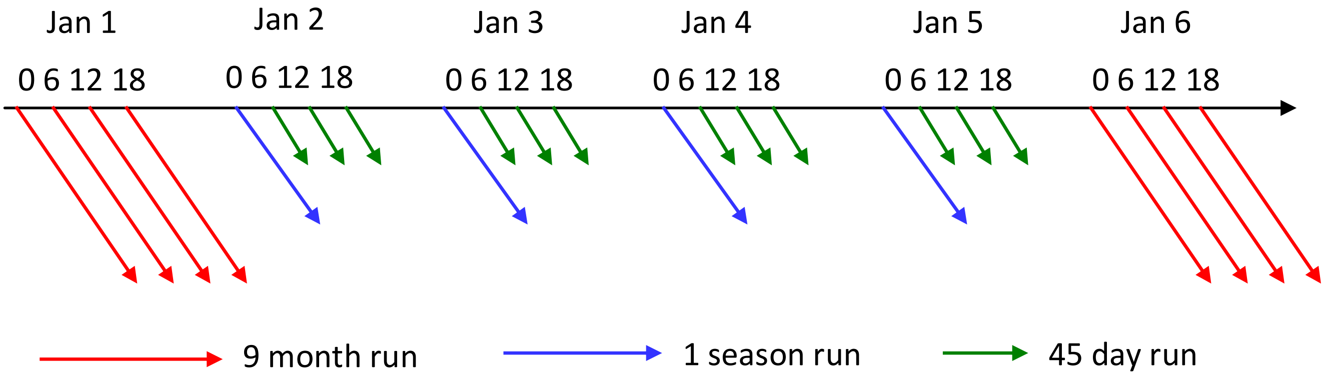


Figure 2.3: CFSv2 Retrospective Forecasts

2.3.3 Operational Configuration

The initial conditions for the CFSv2 retrospective forecasts are obtained from the CFSR, while the real time operational forecasts obtain their initial conditions from the real time operational CDASv2. Great care was made to unify the CFSR and CDASv2 in terms of cutoff times for data input to the atmosphere, ocean and land surface components in the data assimilation system. Therefore, there is greater utility of the new system, as compared to CFSv1 (which had a lag of a few days), since the CFSv2 initial conditions are made completely in real time. This makes it possible to use them for the sub-seasonal (week1-week6) forecasts. Operational real time data may be downloaded from the official site.

There are 4 control runs per day from the 0, 6, 12 and 18 UTC cycles of the CFS real-time data assimilation system, out to 9 months. In addition to the control run of 9 months at the 0 UTC cycle, there are 3 additional runs, out to one season. In addition to the control run of 9 months at the 6, 12 and 18 UTC cycles, there are 3 additional runs, out to 45 days. There are a total of 16 CFS runs every day, of which 4 runs go out to 9 months, 3 runs go out to 1 season and 9 runs go out to 45 days.

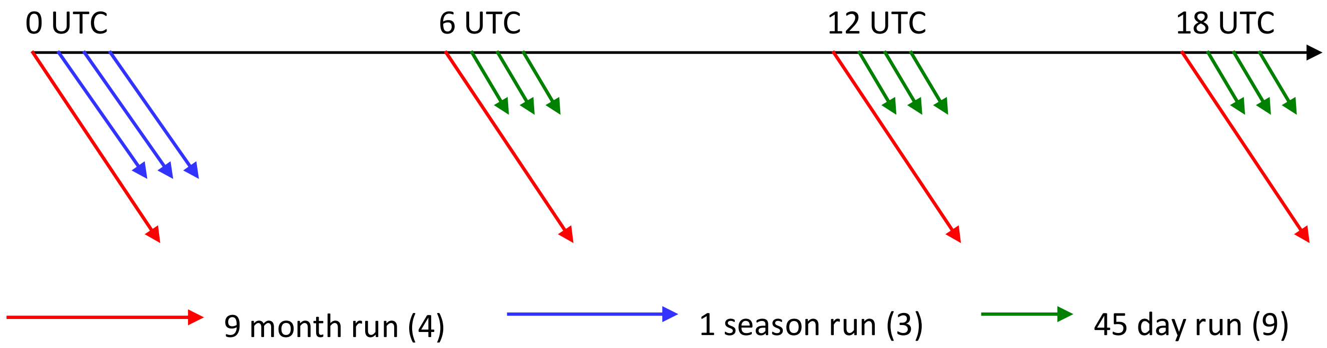


Figure 2.4: CFSv2 Operational Forecasts

The CFSv2 retrospective dataset can be obtained from the official site of CFS.

3. Model Specifications

3.1 Directory Tree

CFSv2 suite has different directories, each intended for a special purpose. The 'com' directory (created at run-time) is the working directory where the model runs and dumps the output.

Following is the directory tree of the model suite:

cfsv2/ (Main directory for the model)

- **bin/** (These scripts control the flow of an experiment)
- **build/** (This directory contains the model installer and the compiler options files)
- **exec/** (This directory contains all the executables for model components)
- **exp/** (This directory typically contains configuration files, rlists and submit script)
- **fix/** (Fix files for different model components)
- **init/** (This directory contains the initial conditions to start the model)
- **jobs/** (These scripts, combined with variable definitions set in configuration, call the main driver scripts)
- **libs/** (This directory contains the model libraries and the corresponding sources)
- **parms/** (This directory contains the control parameters files for model components)
- **scripts/** (Development versions of the main driver scripts)
- **sorc/** (Source directories for all model parts)
- **ush/** (Additional scripts to invoke model components, typically called from within the main driver scripts)
- **util/** (This directory contains utility scripts and executables for running the model)
- **com/** (Working and Output directory)

3.2 Functionality of model components and scripts

3.2.1 Source Modules

1. Dir: global_chgres.fd Exec: global_chgres

This program changes the resolution of the sigma and surface restart files from the global spectral model. The input files should have header records identifying their respective resolutions. The output file's resolution is specified in the namelist file *namchg*. Either the input sigma or surface file may be missing, in which case no counterpart file is created with the new resolution.

The procedure for changing the sigma file resolution is thus:

A new orography is optionally read in. If it is missing, the new orography will be the transform of the old orography. A new sigma structure is also read in. This file is only optional if the number of levels are the same, in which case the new sigma structure defaults to the old sigma structure. Then, the input spectral fields are read in and transformed to the new Gaussian grid. A new surface pressure is calculated hydrostatically based on the new orography. Then the upper-air fields are vertically interpolated to the inferred new pressures. The vertical interpolation is generally cubic Lagrangian in log pressure with a monotonic condition that a new value cannot exceed the range of its immediate old neighbors. Interpolation is linear between the two outer intervals of the old domain. Fields are held constant outside the old domain, except for temperature and humidity below the old domain, where the temperature lapse rate is held fixed at -6.5 k/km and the relative humidity is also held fixed. Finally, all fields are transformed to the new spectral space and written out. Note that all tracers are interpolated unless requested otherwise. Alternatively, if no transforms are needed, then no new orography or sigma structure is read in and the spectral coefficients are directly padded or truncated. Furthermore, if ozone is requested in the output file but is not in the input file, then ozone is generated from climatology and, optionally, a total ozone grib field. The last record precipitation is also interpolated if requested.

The procedure for changing the surface file resolution is thus:

Nearest neighbor interpolation is performed so that land/non-land points on the input grid are mapped to land/non-land points on the target grid. If the input file contains land-ice and the output grid is to have land-ice, then non-land is mapped to non-land, land-ice is mapped to land-ice, ice free land is mapped to ice free land. Optionally, the fields such as albedo, roughness, etc, may be determined on the output grid from sfccycle (which is called from the surface chgres module). The latter is recommended when converting from a low to high resolution grid. A new land-sea mask is optionally read in. If it is missing, the new land-sea mask is interpolated from the old mask. Skin and soil temperature over land are adjusted for differences between the input and output orography. Liquid soil moisture is calculated according to the adjusted temperature. Output orography may be read in from file or interpolated from input orography. Note: Older versions of the surface restart file (before ivs 200501) do not have orography records. In cases where the input surface file is pre 200501, the program will get the orography from the sigma file. Therefore, you must set the options to convert a sigma file as well as a surface file. When changing a pre 200501 file, the program will interpolate only those land fields needed to run the old OSU land model and old sea ice physics. When changing a 200501 file, the program will interpolate/calculate those additional fields needed by the NOAH LSM (max snow albedo, liquid Soil moist, snow depth, precipitation, precipitation type, slope type, max/min greenness) and the new sea ice model (ice depth and fraction). When changing a pre 200501 file to a 200501 file, the program will automatically initialize the above mentioned fields using either guess values or values calculated from sfccycle. The program will also convert from two to four soil layers and vice versa. The program will run on the full or reduced grid depending on the lonsperlat record of the input file or whether the user specifies an external lonsperlat file. The program will initialize all land states for the land-ice physics if desired. The program will scale total soil moisture for any differences in soil type between the input and output grids.

2. Dir: ncep_post_new.fd Exec: ncep_post

This module takes model output in Gaussian (native) grid and writes them out in GRIB format. It interpolates data from model to model surfaces, ISOBARIC PRESSURE surfaces, Above Ground Level (AGL) height surfaces, and THETA & PV surfaces. Currently, the post processor outputs 405 fields (RQSFLD.f). Sample fields generated by the post-processor are Surface related fields, Sounding and Cloud related fields, Fixed fields, Radiance and Brightness fields, Sea Level Pressure and other miscellaneous fields viz. tropopause level fields, FD (Upper Winds, Wind and Temperature Aloft Forecast) level fields, Freezing level height and Relative Humidity, Boundary layer fields, and LFM (Limited-area Fine Mesh Model) and NGM (Nested-Grid Model) look-alike fields.

3. Dir: cfs_ao_coupler.fd **Exec: cfs_mlc_coupler**

This module co-ordinates the execution of the ATMOS and OCEAN models and transmits data between them. The data transmission includes receiving SST from OCEAN and sending to ATMOS, and receiving FLUX from ATMOS and sending to OCEAN.

4. Dir: cfs_cdas_atmos_fcst.fd **Exec: cfs_cdas_atmos_fcst**

This module is the atmospheric component (GFS) of CFSv2 with triangular truncation T126 (~0.937°) for horizontal resolution and 64 hybrid sigma-3 pressure levels in the vertical; finite difference method is used in the solution of the equations. This module will be coupled with Ocean Model using a coupler module. The atmosphere model (AM), the Ocean Model (OM) and the coupler will be run simultaneously in MPMD fashion. This module can run in coupled mode as well as stand-alone mode. The coupling frequency is flexible up to the Ocean Model time step.

5. Dir: cfs_mppnccombine.cd **Exec: cfs_mppnccombine**

This module joins together NetCDF data files representing a decomposed domain into a unified NetCDF file. It was originally designed to be used as a post processor for the parallel I/O programming interface "mpp_io_mod". If the user is running the source code on one processor, the domain is not decomposed and there is only one data file. *mppnccombine* requires decomposed dimensions in each file to have a 'domain_decomposition' attribute. This attribute contains four integer values: starting value of the entire non-decomposed dimension range (usually 1), ending value of the entire non-decomposed dimension range, starting value of the current chunk's dimension range and ending value of the current chunk's dimension range. *mppnccombine* also requires each file to have a 'NumFilesInSet' global attribute which contains a single integer value representing the total number of chunks (i.e., files) to combine.

6. Dir: cfs_ocean_mom4ice.fd Exec: cfs_ocean_mom4ice

The Modular Ocean Model (MOM) is a numerical representation of the ocean's hydrostatic primitive equations. It is designed primarily as a tool for studying the global ocean climate system, but with recent enhanced capabilities for regional and coastal applications. As with all previous versions of MOM, MOM4 discretizes the ocean’s hydrostatic primitive equations on a fixed Eulerian grid, with the Arakawa B-grid defining the horizontal arrangement of model fields. That is, the grid cells live on a lattice fixed in space-time. Given that, MOM4 remains a z-coordinate ocean model.

MOM4 has been coded within GFDL's Flexible Modeling System (FMS). Doing so allows for MOM4 developers to use numerous FMS infrastructure and superstructure modules that are shared amongst various atmospheric, ocean, sea ice, land, vegetative, etc. models. The following list represents a sample of the FMS 'shared' modules used by MOM4.

1. *time manager:* keeps time and sets time dependent flags
2. *coupler and data override:* used to couple MOM4 to other component models and/or datasets.
3. *I/O:* to read and write data
4. *initial and boundary data:* regrids spherical fields to the generally non-spherical ocean model grid
5. *grid and topography specification:* sets model grid spacing and interpolates spherical topography to the model grid
6. *parallelization tools:* for passing messages across parallel processors
7. *diagnostic manager:* to register and send fields to be written to a file for later analysis
8. *field manager:* for organizing multiple tracers for use especially in bio-geochemistry studies

The complete model documentation and the MOM4 Manual can be obtained from http://data1.gfdl.noaa.gov/~arl/pubrel/o/old/doc/mom4p0_guide.pdf and http://data1.gfdl.noaa.gov/~arl/pubrel/o/old/doc/mom4_manual.html respectively.

7. Dir: cfs_overparm_grib.fd Exec: cfs_overparm_grib

This module reads an entire GRIB file and writes it back out, replacing the internal parameter table version and parameter id read in. Change is made to that id only if the replacement is positive. The non-GRIB information in the input GRIB file will be lost, if it is present. An output line will be written for each GRIB message.

8. Dir: cfs_psichi.fd Exec: cfs_genpsiandchi

This module reads wind on grid points, transform them back to spectral space, and then computes Stream function (Psi), velocity potential (Chi), vorticity, and divergence.

9. Dir: global_sighdr.fd Exec: global_sighdr

This module extracts and prints the information from the sigma header of a sigma file. It can print the parameters such as type of the file (filetype), truncation number (jcap), number of vertical coordinates (nvcoord). Running sfchdr with no additional arguments (other than the input file) as in the last example allows for keyboard input of multiple variables, one at a time, until the program is interrupted (eg, via CTRL-c). Enter "?" (without the quotes) as standard input and then all the possible input values will be printed.

10. Dir: cfs_tripo2reg.fd Exec: cfs_tripo2reg

This module reads daily ocean and ice data on tripolar grid and interpolates the data onto regular lat/lon grid.

11. Dir: cfs_separ3.fd Exec: cfs_separ3

This module separates the output log of the parallel coupled run into three independent files, one each for atmosphere, ocean and coupler.

3.2.2 Utilities

Name	Despcription
anomgb	This module reads all or part of one GRIB file, computes climate anomalies, and writes the anomalies to another GRIB file, interpolating if necessary. Only geopotential height anomalies are computed in the current implementation. Unless otherwise directed (-x option), the GRIB index file is also used to speed up the reading. The fields are interpolated to an output grid if specified (-g option). The interpolation type defaults to bilinear but may be specified directly (-i option). The copying may be limited to specific fields (-k option). The command may be directed to output verbose diagnostics (-X option). If grib2 is '-', the output GRIB file is written to standard output.
copygb	Copygb copies all or part of one GRIB file to another GRIB file, interpolating if necessary. Unless otherwise directed (-x option), the GRIB index file is also used to speed up the reading. The fields are interpolated to an output grid if specified (-g option). The interpolation type defaults to bilinear but may be specified directly (-i option). The copying may be limited to specific fields (-k option). It may also be limited to a specified subgrid of the output grid or to a subrange of the input fields (-B and -b, -A, and -K options). Fields can be identified as scalars or vectors (-v option), which are interpolated differently. The invalid data in the output field can be filled with mask values or merged with a merge field (-M and -m options). The output GRIB message can also be appended to a file (-a option). If grib2 is specified as '-', then the output GRIB file is written to standard output.
grbindex	It creates an index file from a GRIB file. The index file serves as a table of contents for the GRIB file, enabling quick access to the data. The GRIB file must be unblocked, but there can be a gap before the first GRIB message of at most 32000 bytes and gaps between messages of at most 4000 bytes. The two file names are retrieved from the command line arguments. The first argument is the name of the input GRIB file. The second argument is the name of the output index file. Currently, only version 1 of GRIB can be read.
wgrib	The command wgrib both inventories and decodes GRIB-1 files. There are three types of inventories (regular, short, and verbose) which can be viewed as a

	human-readable index file. The inventories can be manipulated to select the records to decode. The output formats of wgrib include: text, binary (system dependent), big endian IEEE and GRIB. In addition the program can produce a non-inventory description of the GRIB records. Information includes range of values, grid type, etc. The program can be compiled to either use the NCEP operational GRIB tables or the NCEP/NCAR Reanalysis GRIB table as the default table in cases of ambiguity. The program does not handle spectral files nor files with complex packing.
ndate	Module to compute verifying date, given the forecast hour and the initial date. The forecast hour may also be negative and the output verifying date and the initial date are in YYYYMMDDHH format.
nhour	Module to compute forecast hour given the verifying date and the initial date. The verifying date and the initial date are in YYYYMMDDHH format. Leading zeros are added to make forecast hour at least two digits and a leading minus in the forecast hour signifies that the initial date comes after verifying date.

3.2.3 Scripts

i) util/ush/

Name	**Description**
errexit.sh	This script is to be used when a fatal error or condition has been reached and you want to terminate the job. The script puts fail messages in job output file and in the job logfile, a fail message is sent to the front end, and processing is terminated.
finddate.sh	This script looks in either forward or backward in time to generate either a variable containing sequential date/time stamps for a period up to a month or just the date/time stamp occurring at the end of such a period. Time stamp is in the form yyyyddmm. Leap years are accounted for.
postmsg.sh	This script posts messages to a log file. The script assumes that the variable

	"jobid" has been exported by the parent shell. The variable "jobid" contains the name of the job and its process id in the form "jobname.pid."
startmsg.sh	This script posts the 'program started' message to the log file when any program starts.

ii) ush/

Name	Description
global_anomcat.sh	This script computes the height anomalies at 1000 and 500 mb and the five-wave height anomaly at 500 mb and concatenates them to a pressure GRIB file.
global_chgres.sh	This script changes the resolution of the global restart files, namely the sigma file or the surface file or both. The resolution of the output files is given in the argument list or as imported environment variables. The resolution of the input files are taken from the header records in the respective files. Resolution is given as spectral truncation, number of levels, number of longitudes and number of latitudes.
global_nceppost.sh	This script reads a single global GFS IO file and (optionally) a global flux file and creates a global pressure GRIB file. The resolution and generating code of the output GRIB file can also be set in the argument list.
reconcile.sh	This script sets the final environment for the forecast after the basic environment has been set in para_config file. It sets required, but unset variables to default values. With this version forecasts can be made using two model resolutions.
post_mdl.sh	This script runs the post processor for the forecast.

iii) scripts/

Name	Description
excfs_cdas_fcst.sh.s	This is a MPMD coupled script to run CFS using the GFS script for

ms	Atmospheric Model and MOM4 script for Ocean Model. Atmosphere model part of this script runs a global spectral atmosphere model. The initial conditions and run parameters are passed in the argument list. The Ocean model part runs the Modular Oceanic Model 4 for the number of days specified in the namelist "namelist.control". The script sets up the pre-execution environment for the atmospheric, oceanic and coupler components and also performs the post-execution tasks (saving output files, appending date suffix to filenames for identification, removing intermediate files) after the completion of the model run.

iv) jobs/

Name	**Description**
fcst.sh	This script is the main driver script to run the forecast. This script sets the model variables along with variables for MPMD execution. The script runs CHGRES (optionally) before the forecast.
ocnp.sh	This script runs the post processor for the ocean model. The daily ocean and ice data in NETCDF format is interpolated and written in GRIB1 format.
post.sh	This script runs the post processor for the atmosphere model. The native model output in binary/IEEE/GRIB format is written to GRIB1 format.

v) bin/

Name	**Description**
ncpx	This command efficiently copies files, particularly for NFS.
pbeg	This script runs when parallel jobs begin.
pend	This script runs when parallel jobs end. It sets the sequence of execution of job steps.
perr	This script runs when parallel jobs fail.

pcne	This script counts non-existent files.
plog	This job logs parallel jobs.
pmkr	This script makes the *rlist*, the list of data flow for the experiment.
pcon	This script searches the input (rlist) for patterns and returns the assigned value.
psub	This script checks the pre-requisites for the job steps and runs parallel jobs.

vi) exp/

Name	**Description**
para_config	This is the configuration file for CFS or GFS. It sets the options for the model components. Various model configuration options such as coupled/stand-alone case, post options, paths for source scripts/output directories, utilities, computing nodes, output resolution, etc. can be set here.
submit.sh	This is the start script i.e. the first script to be executed. It submits the forecast job.
pbs.submit	PBS script to submit the job on Linux cluster.
create_tar.sh	Script to create a tar bundle of model output.

3.2.4 Libraries

Name	**Description**
bacio	This library is responsible for performing byte addressable input and output operations. It contains the FORTRAN-callable routines to read and write character (bacio) and numeric (banio) data byte address-ably. It includes the byte addressable read and write operations on the file descriptor that is provided.
crtm	**Community Radiative Transfer Model** This library contains several routines that involve gathering of sensor data and computing derived geometry, infrared sea surface emissivity (IRSSE) for input wind speed, frequency, and angles, surface emissivity and reflectivity at infrared

	frequencies, surface optical properties, tangent-linear surface emissivity and reflectivity at infrared frequencies for land surface, water surface, snow surface, and ice surface.
esmf	**Earth System Modeling Framework** The Earth System Modeling Framework (ESMF) collaboration is building high-performance, flexible software infrastructure to increase ease of use, performance portability, interoperability, and reuse in climate, numerical weather prediction, data assimilation, and other Earth science applications. The ESMF defines an architecture for composing complex, coupled modeling systems and includes data structures and utilities for developing individual models. The ESMF library is responsible for representing large scale physical domains i.e. atmosphere and ocean components. It creates a coupler that intermediates the data between these components. Grid interpolation and re-mapping are core utilities of ESMF. It also creates an ESMF Grid object and sets up its internal structure so that it is usable for other Grid methods. ESMF takes care of reading the coordinates to form appropriate grids. Re-gridding is based on bi-linear or bi-cubic interpolation, conservative re-mapping, spectral and other functional transforms**.** ESMF also includes toolkits for building components and applications, such as regridding software, calendar management, logging and error handling, and parallel communications.
gfsio	This library takes model output in Gaussian (native) grid and writes them out in GRIB format.
ip	The general interpolation library “ip” contains FORTRAN subprograms to be used for interpolating between almost any grids used at NCEP. There are currently five interpolation methods available in the library. They are respectively bi-linear, bi-cubic, neighbor, budget and spectral interpolation methods. Generally, only regular grids can be interpolated in this library.

landsfcutil	This library interpolates data to the model grid by taking the nearest neighbor or area average of the source data and interpolates data to the model grid by bi-linear interpolation. Contains collection of routines that go from lat/lon to x/y space on various grids and collection of routines that perform soil/land related calculations such as roughness length, soil type specific parameters, albedo based on snow water equivalent, snow-free albedo, maximum snow albedo, liquid portion of total soil moisture, calculate super-cooled soil moisture and re-scale total soil moisture for a change in soil type.
nam_nmm_fcst_real	**WRF-NMM V2.1** The WRF-NMM model is a fully compressible, non-hydrostatic model with a hydrostatic option. The terrain following hybrid pressure sigma vertical coordinate is used. The grid staggering is the Arakawa E-grid. The same time step is used for all terms. Time stepping used are: Horizontally propagating fast-waves: Forward-backward scheme, Vertically propagating sound waves: Implicit scheme. Forward, second order "Smagorinsky-type" horizontal diffusion is used. And for the vertical diffusion free atmosphere turbulence above surface layer is used.
netcdf	NetCDF (network Common Data Form) is a set of interfaces for array-oriented data access and a freely distributed collection of data access libraries for C, Fortran, C++, Java, and other languages. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data. NetCDF files are self-describing, network-transparent, directly accessible, and extendible. ● **Self-describing:** It means that a NetCDF file includes information about the data it contains. ● **Network-transparent:** It means that a NetCDF file is represented in a form that can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.

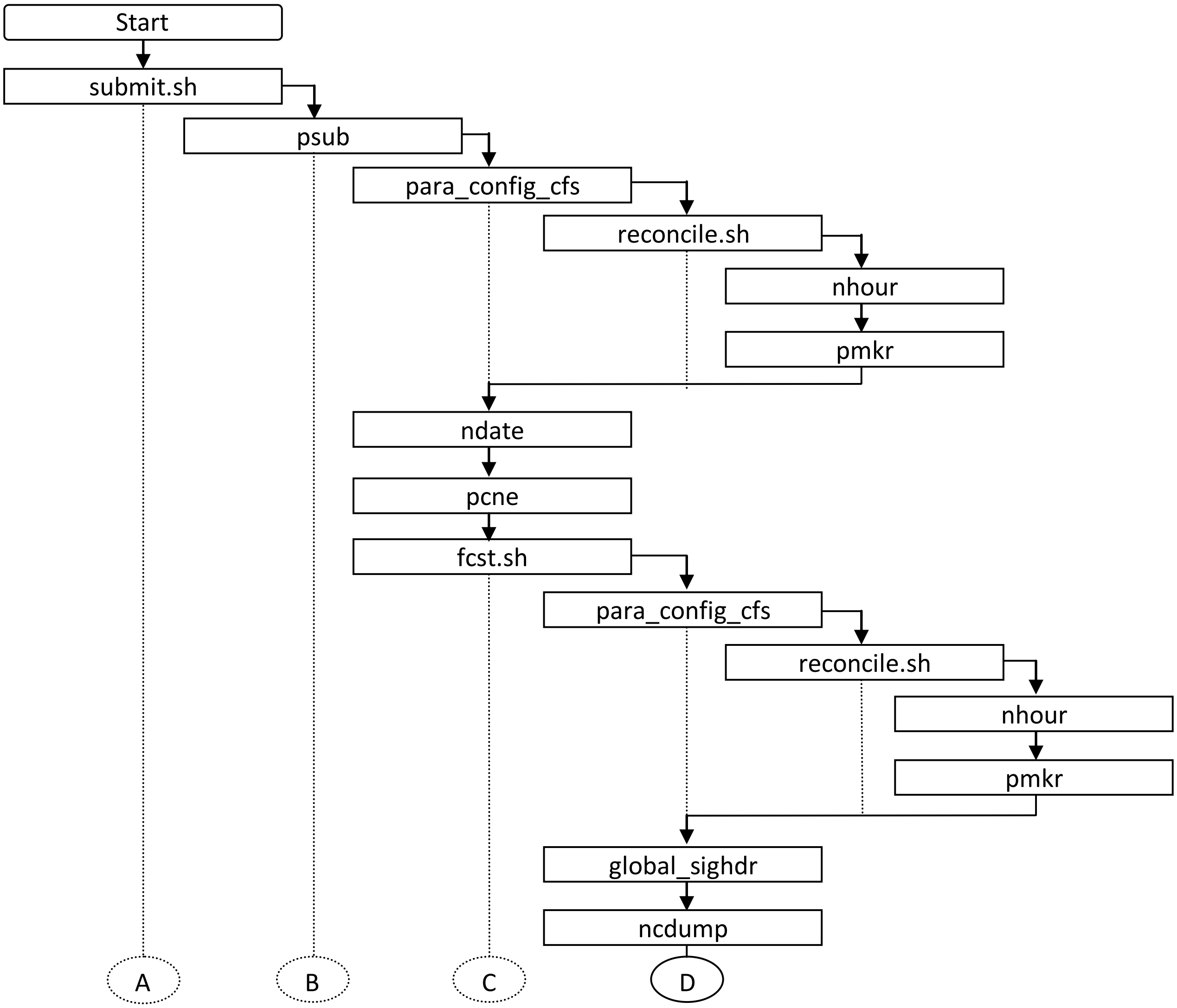
	● **Direct-access:** It means that a small subset of a large data-set may be accessed efficiently, without first reading through all the preceding data. ● **Extendible:** It means that data can be appended to a NetCDF data-set without copying it or redefining its structure. NetCDF files are used for creating oceanic output files for platform independent usability.
sfcio	This library provides an application Program Interface for performing I/O on the surface restart file of the global spectral model. Functions include opening, reading, writing, and closing as well as allocating and de-allocating data buffers used in the transfers. The I/O performed here is sequential. The transfers are limited to header records or data records.
sigio	This module provides an Application Program Interface for performing I/O on the sigma restart file of the global spectral model. Functions include opening, reading, writing, and closing as well as allocating and de-allocating data buffers used in the transfers. The I/O performed here is sequential and random. The transfers are limited to header records, data records, surface data records, or specific levels of upper air data records.
sp	Analyzes spectral coefficients from Fourier coefficients for a latitude pair. This library spectrally truncates vector fields on a global cylindrical grid, returning the fields to a possibly different global cylindrical grid. The wave-space can be either triangular or rhomboidal. Either grid-space can be an equally-spaced grid or a Gaussian grid. It performs multiple Fast Fourier Transforms between complex amplitudes in Fourier space and real values in cyclic physical space. It is used for performing various operations in spectral space.
w3	This library is used to unpack & read GRIB files and also pack & write GRIB files. The several modules in this library are used for identifying the size of various sections of the GRIB file and reading the appropriate number of bytes. It also performs the conversion from ASCII to EBCDIC, or from EBCDIC to ASCII by character.

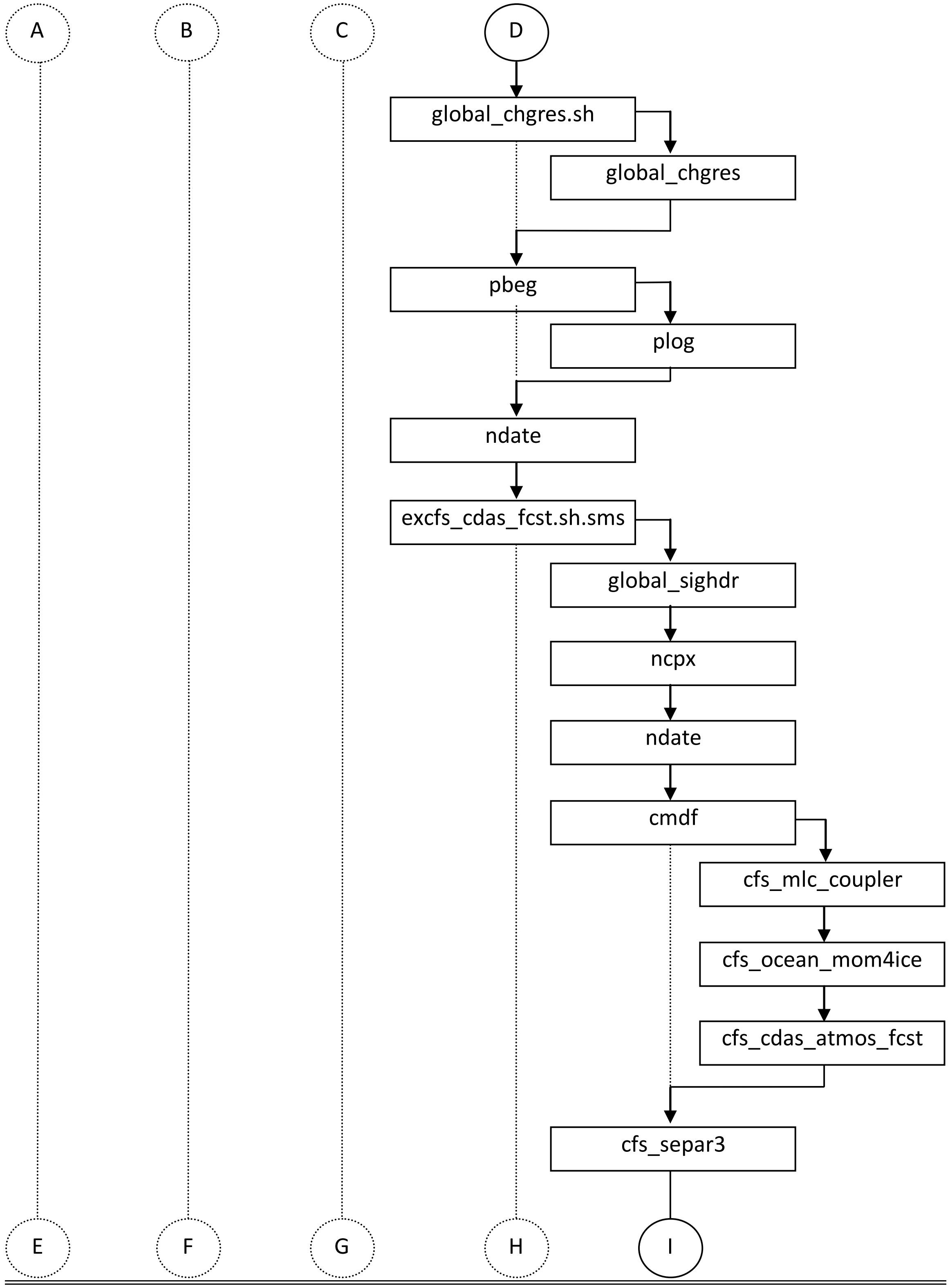
3.3 Call flow of the model

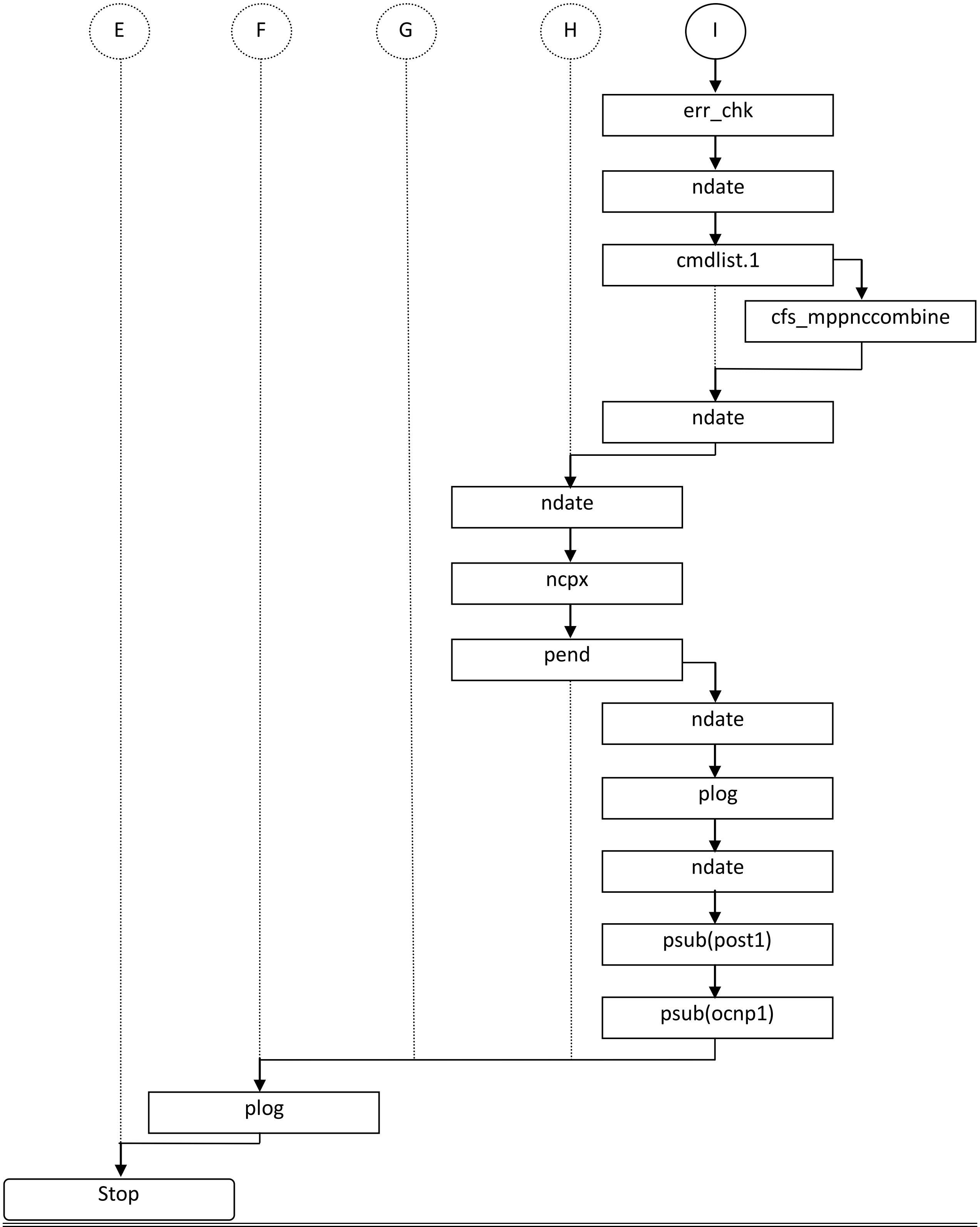
The model enters the initialization phase before entering into to the forecast phase. All model executables are called in the respective shell scripts, which first initialize the global variables required for the model run. These variables are used throughout the model code. Certain variables pertaining to machine path can be changed as per the user convention. This is explained later in section 4.3.

3.3.1 Call flow of the forecast program

Following tree describes the flow of code of the CFS model.



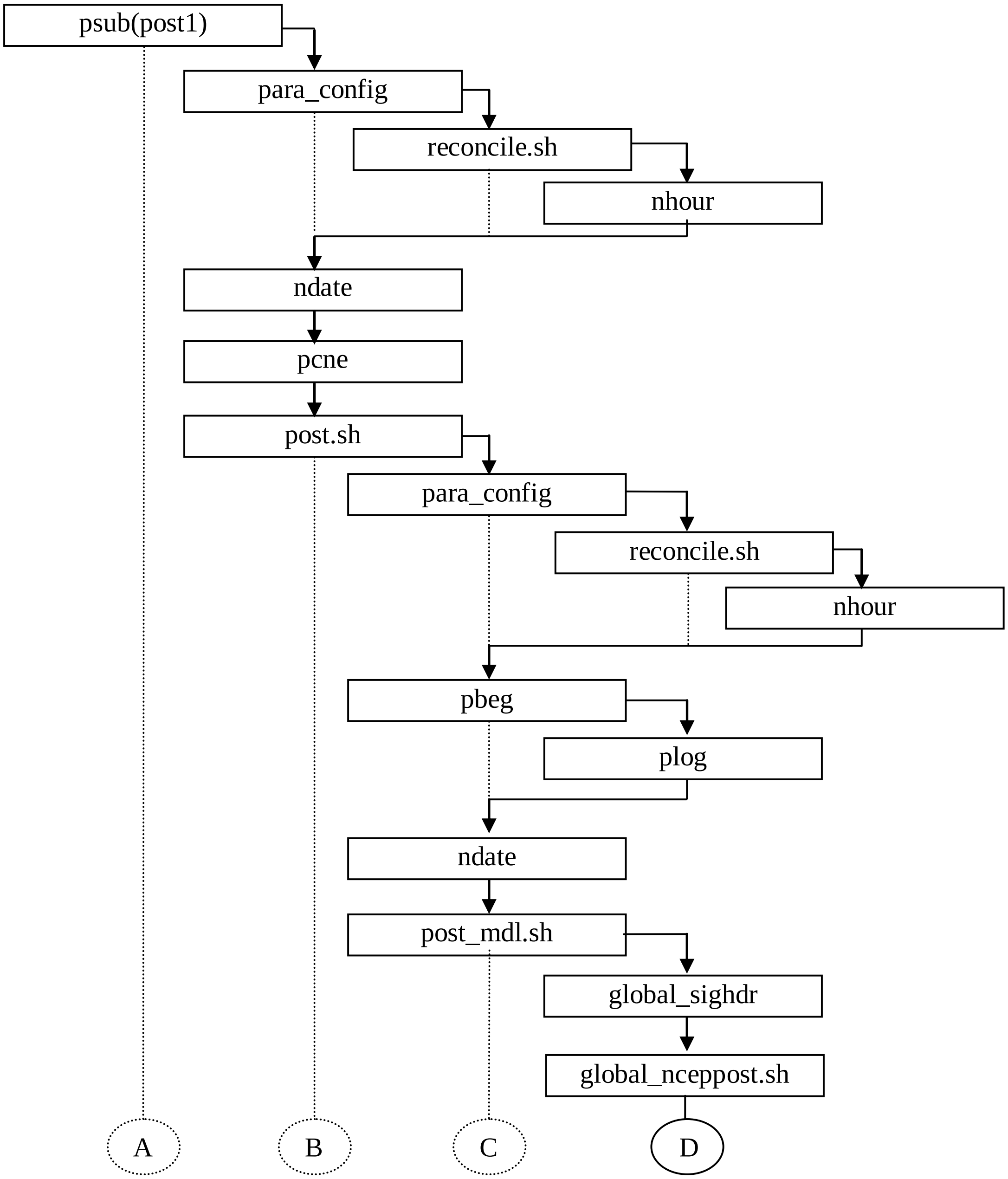

A
B
C
D
global_chgres.sh
global_chgres
pbeg
plog
ndate
excfs_cdas_fcst.sh.sms
global_sighdr
ncpx
ndate
cmdf
cfs_mlc_coupler
cfs_ocean_mom4ice
cfs_cdas_atmos_fcst
cfs_separ3
E
F
G
H
I

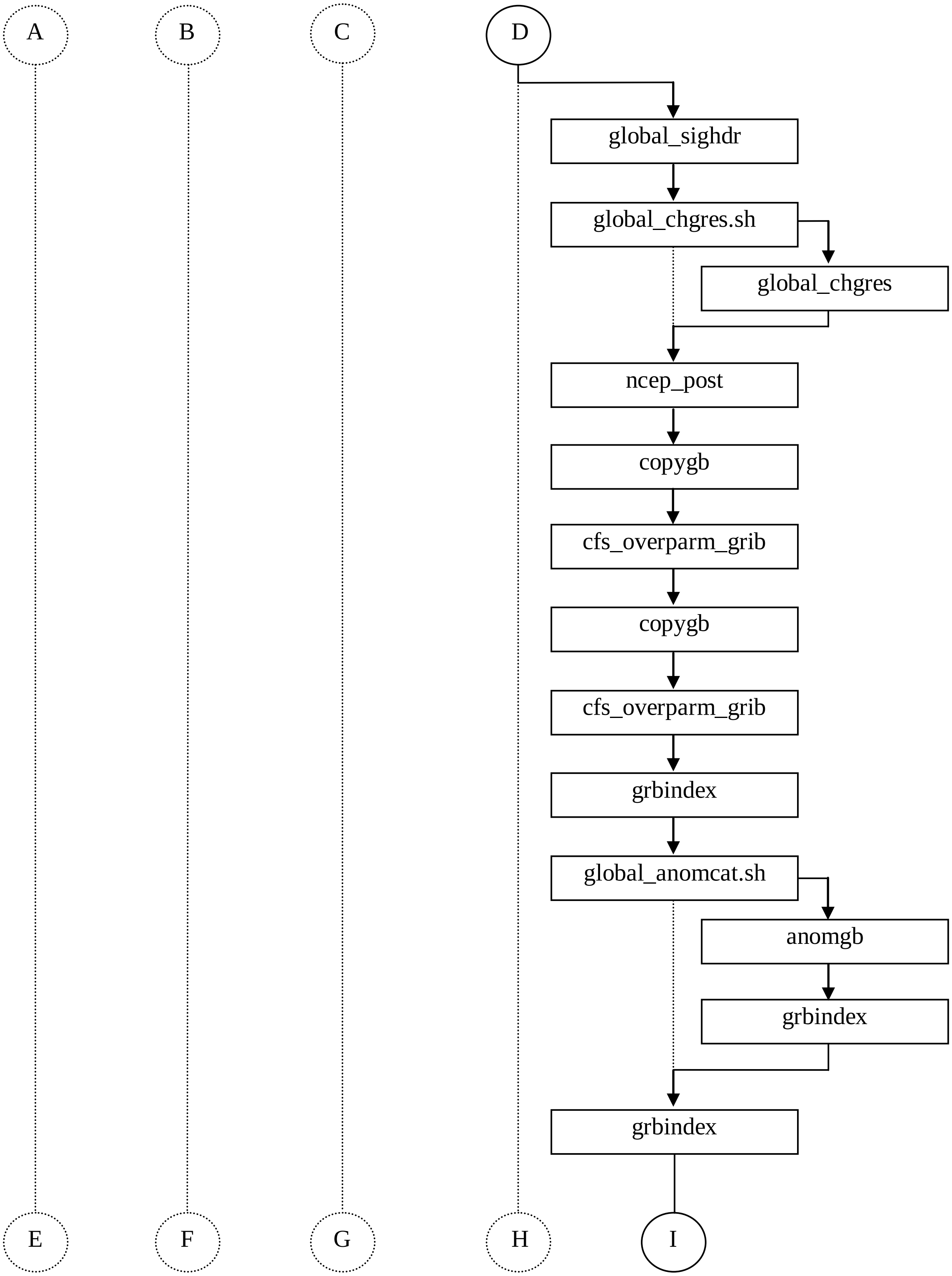

E
F
G
H
I
err_chk
ndate
cmdlist.1
cfs_mppnccombine
ndate
ndate
ncpx
pend
ndate
plog
ndate
psub(post1)
psub(ocnp1)
plog
Stop

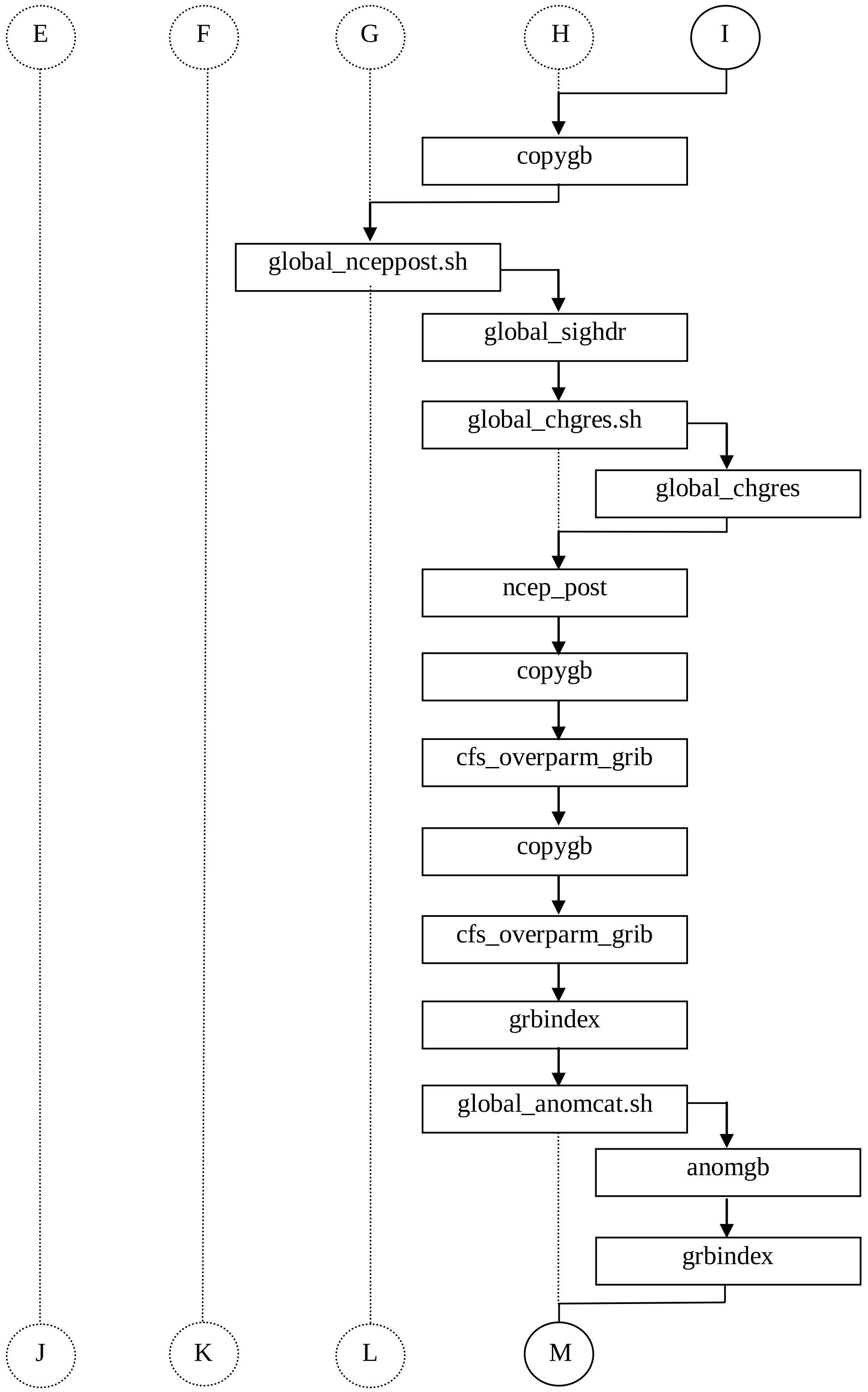
3.3.2 Call flow of the post-processing program

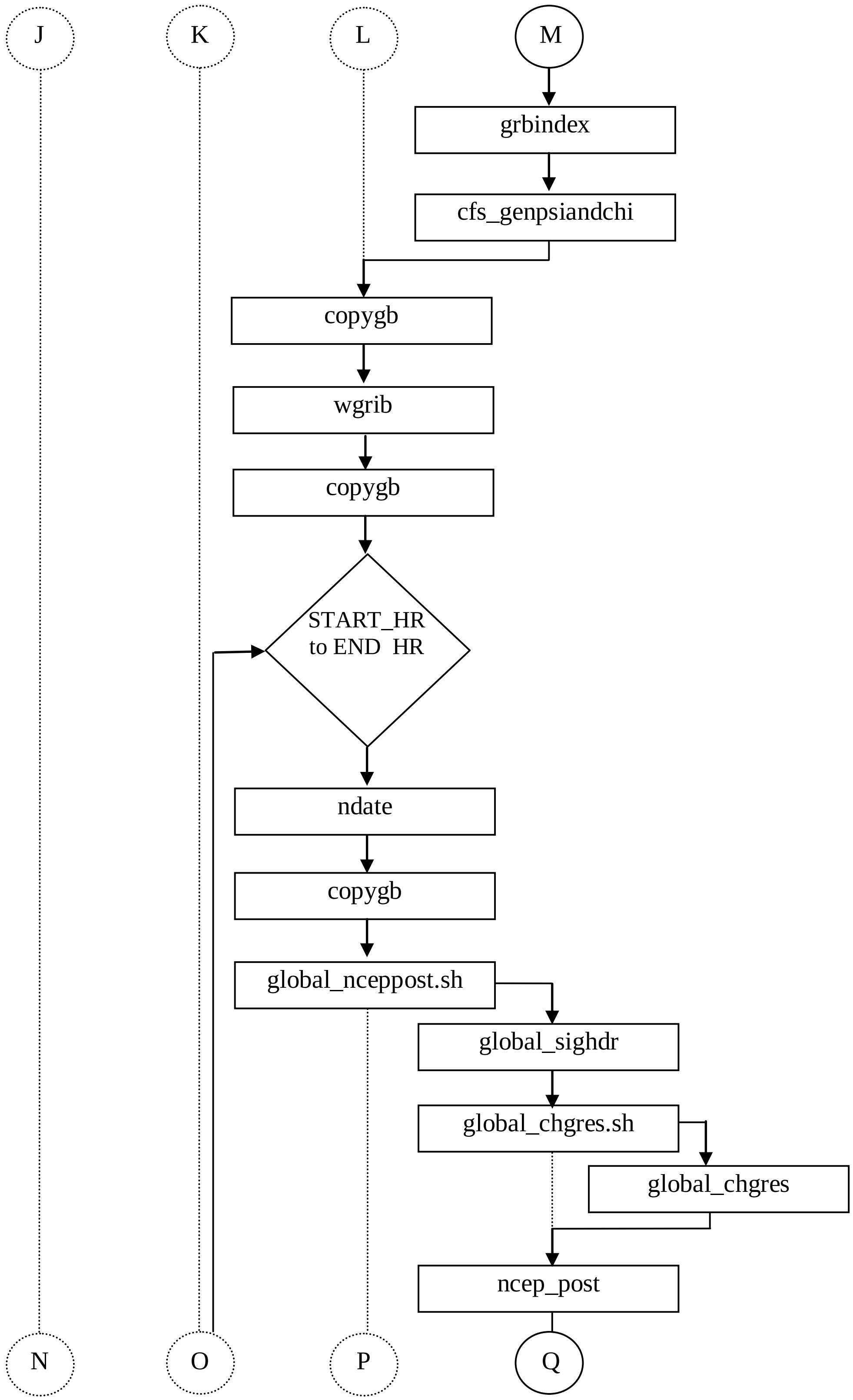
The post-processor follows the same pattern for initialization as followed by the model forecast program. Post-processing scripts are called after initialization, which in turn call the respective executable files.

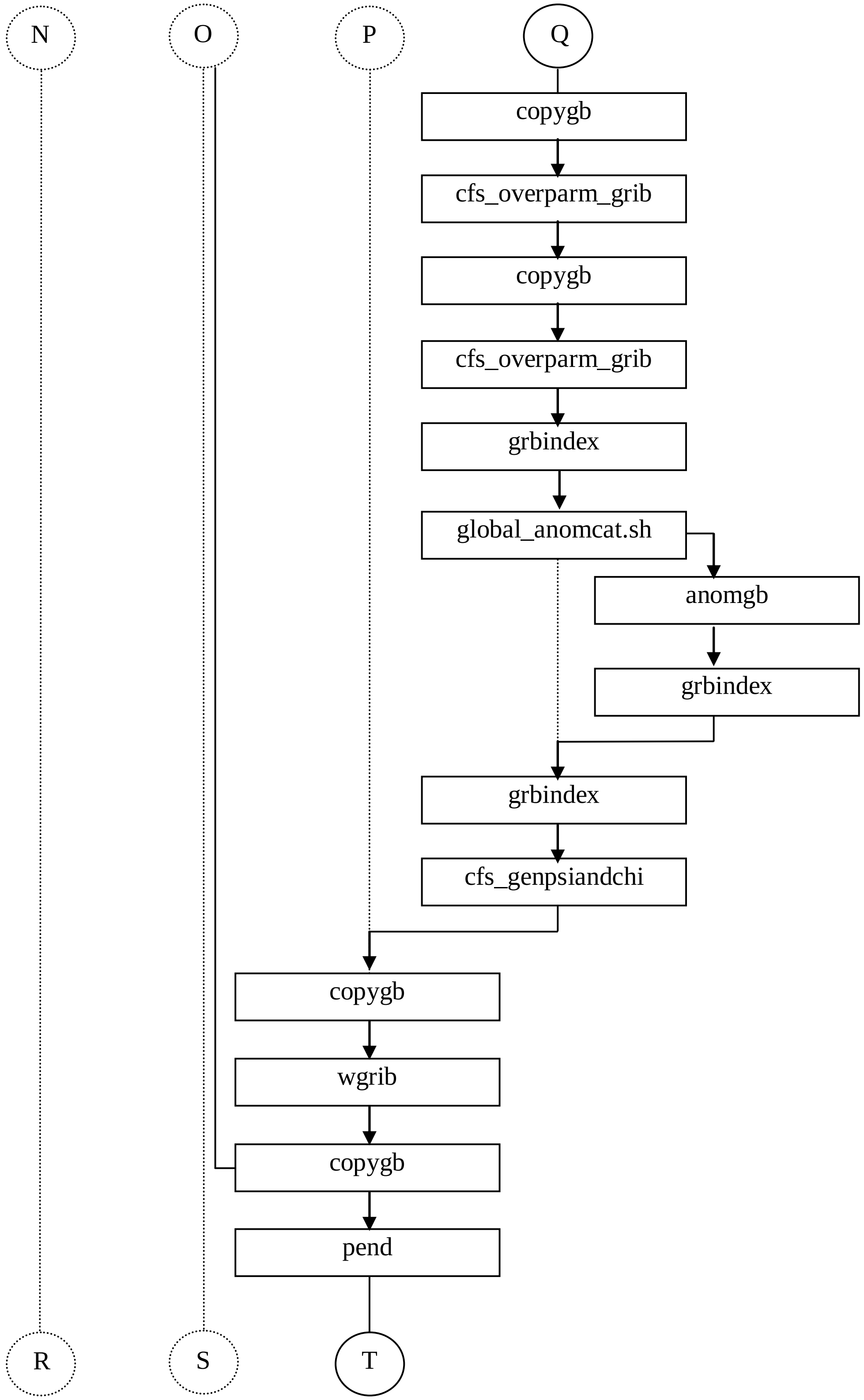
3.3.2.1 Post-processor for the Atmosphere

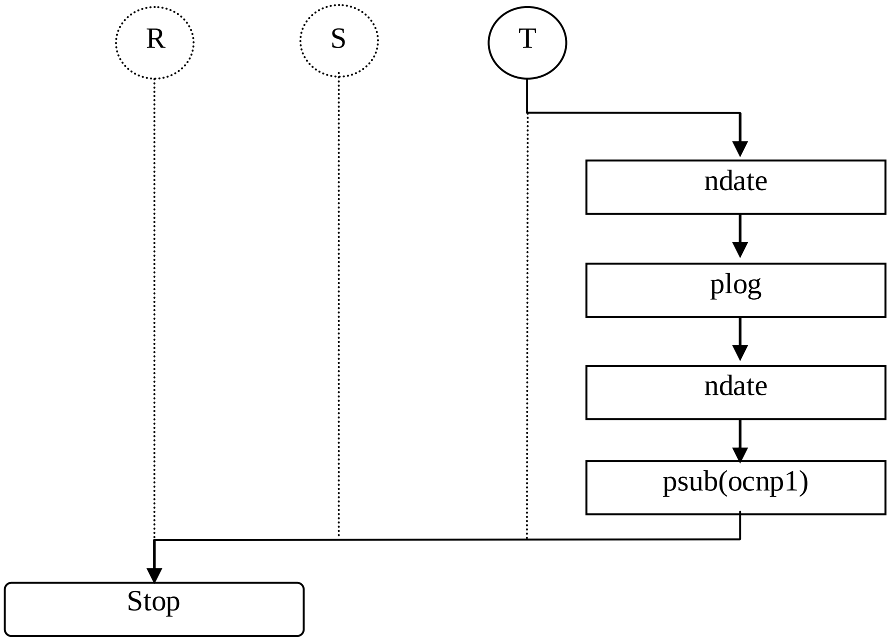



A
B
C
D
global_sighdr
global_chgres.sh
global_chgres
ncep_post
copygb
cfs_overparm_grib
copygb
cfs_overparm_grib
grbindex
global_anomcat.sh
anomgb
grbindex
grbindex
E
F
G
H
I

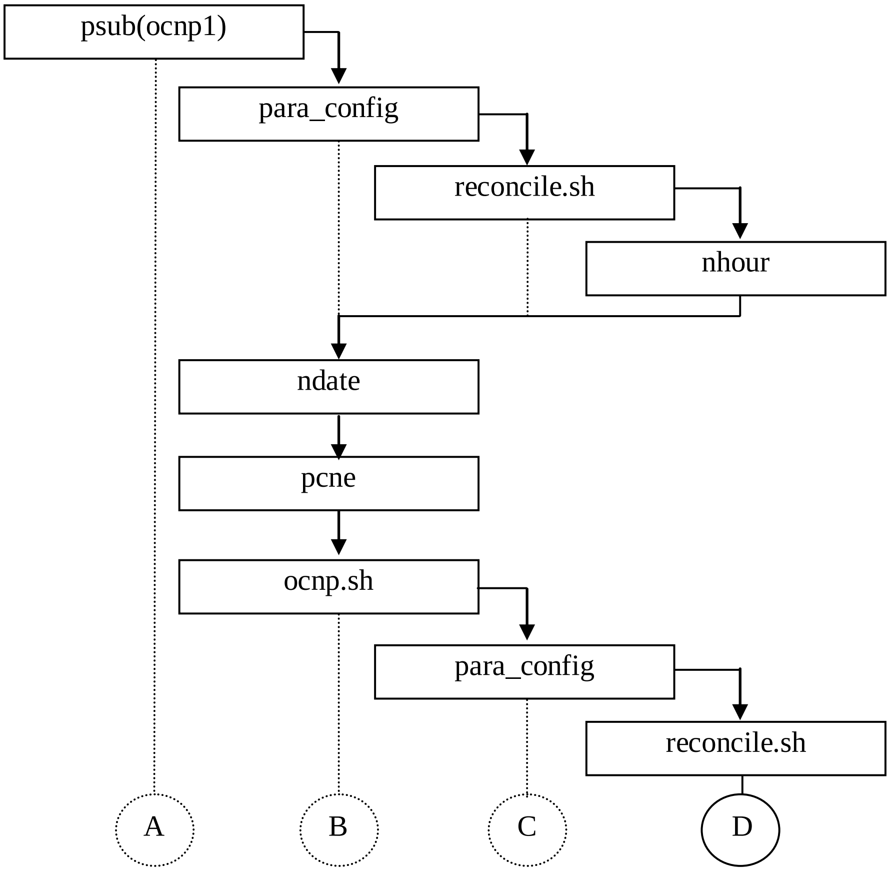

E
F
G
H
I
copygb
global_nceppost.sh
global_sighdr
global_chgres.sh
global_chgres
ncep_post
copygb
cfs_overparm_grib
copygb
cfs_overparm_grib
grbindex
global_anomcat.sh
anomgb
grbindex
J
K
L
M

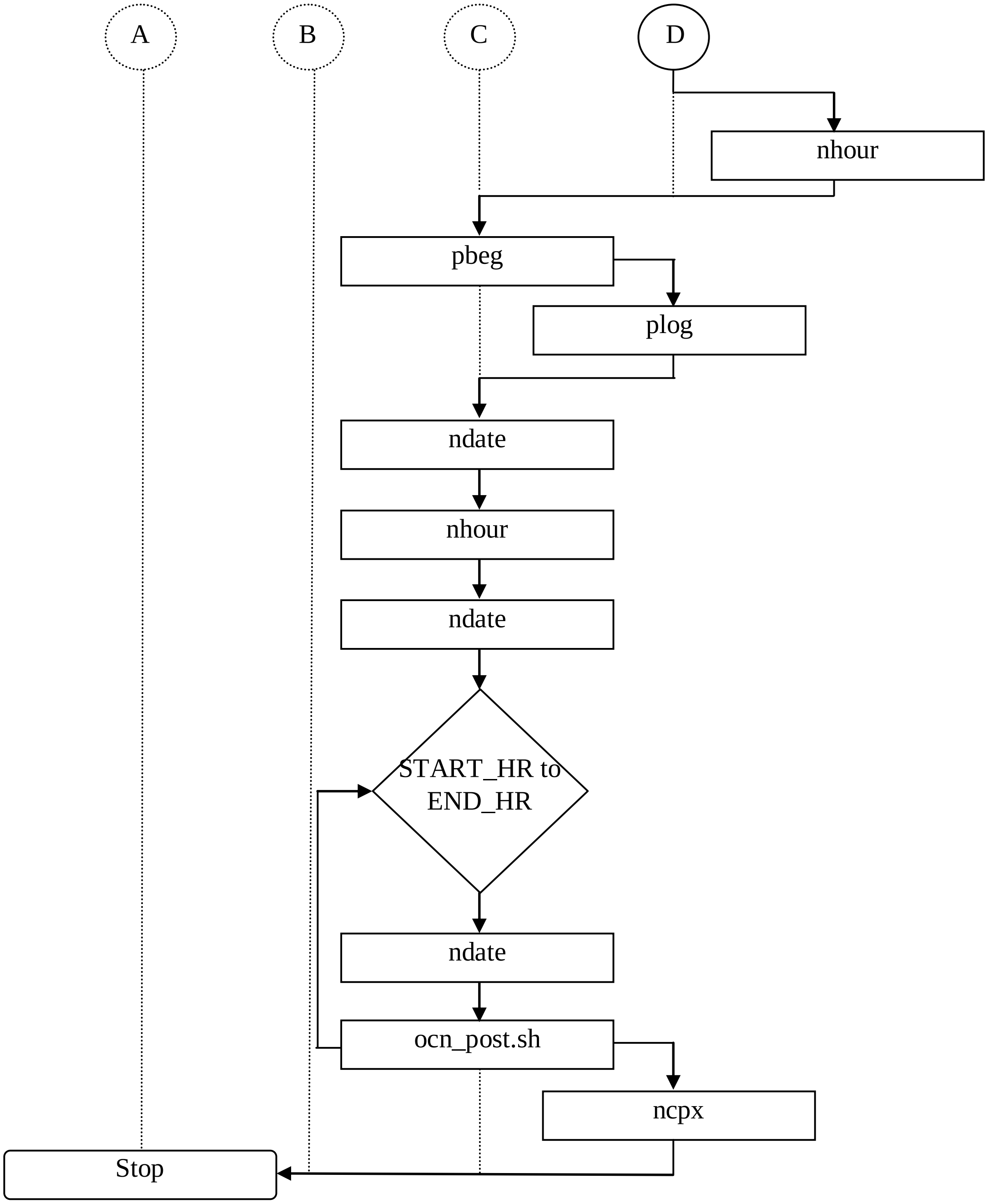

J
K
L
M
grbindex
cfs_genpsiandchi
copygb
wgrib
copygb
START_HR
to END HR
ndate
copygb
global_nceppost.sh
global_sighdr
global_chgres.sh
global_chgres
ncep_post
N
O
P
Q


N
O
P
Q
copygb
cfs_overparm_grib
copygb
cfs_overparm_grib
grbindex
global_anomcat.sh
anomgb
grbindex
grbindex
cfs_genpsiandchi
copygb
wgrib
copygb
pend
R
S
T



3.3.2.2 Post-processor for the Ocean




A
B
C
D
nhour
pbeg
plog
ndate
nhour
ndate
START_HR to END_HR
ndate
ocn_post.sh
ncpx
Stop

4. Installation and Setup

4.1 About CTSF (GG-BLR)

As part of the Garuda Grid Computing Initiative, 4TF Linux cluster was setup at CTSF which provides resources for Grid applications. The GG-BLR cluster consists of 40 Nos. HP Proliant DL160 compute-nodes and 1 No. of HP Proliant DL360 head-node. Each of the compute nodes has Intel Xeon 8 core X5460 processors @3.16GHz and 16 GB memory. It has a total of 320 processors and 640 GB of memory and the nodes are connected by Infiniband interconnect. GG-BLR has a theoretical peak of 4044.80 Gflops/secs (~4TF). Table 4.1 describes the hardware and software configuration of the GG-BLR cluster.

Table 4.1: Hardware and Software configuration of GG-BLR

GG-BLR Cluster

Head node / File server	
Processor	2 X Quad core Xeon @ 3.16 Ghz
Memory	24 GB
Internal Storage	2 / 4 * 146 G SAS HDD
Operating System	Rocks 5.0 on RHEL 5.1 x86_64
Compute Node	
Processor	2 X Quad core Xeon @ 3.16 Ghz
Memory	16 GB
Internal Storage	2 X 250 G SATA HDD
Operating System	Rocks 5.0 on RHEL 5.1 x86_64
Networks	
Primary	Infiniband @ 20 Gbps Full Duplex
Backup	Gigabit Ethernet @ 1 Gbps Full Duplex
Management	10/100 Mbps Fast Ethernet

External Storage

Storage Array	10 TeraBytes SAS 24 TeraBytes SATA

Operating System

RHEL 5.1 x86_64 on Rocks 5.0

Compilers and Related Tools

*Intel Compiler Suite 11.0
*Intel MKL
*MVAPICH2
*MPICH2 / OPENMPI

Local Resource Manager

* Torque 2.3

The GG-BLR cluster is also accessible to users from remote locations via the GARUDA Grid infrastructure.

4.2 Model Installation

The Climate Forecast System has been successfully ported on the HPC facility available at C-DAC, Bangalore i.e. GG-BLR. Since the code is architecture-bound, hence, some modifications in the code and the installation options, specific to the platform, may be needed for successful installation.

The CFS was ported on GG-BLR with MPP configuration. Since the model has been specifically designed for the IBM AIX platform, the source code, model scripts and the compilation options were significantly changed to accommodate the change in platform. While a few modules supplied with the bundle were found to be incomplete, they were replaced with the modules available at the NCEP website. A few missing files were separately downloaded from the website and placed at the respective locations. Installation of CFS on a hardware similar to

GG-BLR should be straight-forward, without much difficulties, except the fact that care must be taken when using different versions of compilers and the supporting libraries.

The process of building and running the Climate Forecast System on a Linux platform is performed in the following stages:

- Building the required libraries for the model.
- Building the source and utility modules to create the corresponding model executables.
- Configuring the model for the run w.r.t. run-length, processor-configuration, output frequencies, etc.
- Submitting the model job to cluster system using the cluster job submission process.

4.2.1 System Requirements

The minimum set of specifications required to build the CFSv2 model on a CDAC-like Linux system are described here. This does not restrict the installation of the model on other hardware/software platforms. The hardware specifications of the cluster where the model has been installed and tested have been mentioned in table 4.1. Following system software are minimally essential for building the model.

Libraries

- Intel Math Kernel Library (MKL) and its dependencies
- Intel MPI and its dependencies
- OpenMP
- NetCDF

Compilers

- Intel FORTRAN and C compilers
    - ifort
    - icc
    - mpicc
    - mpiifort

4.2.2 Installation Instructions

The CFSv2 build mechanism is fairly simple. Three components need to be built for CFSv2 – Model Libraries, Source modules and Utilities. Each of these components follows a two-step installation process – Configuration and Compilation/Building. The first step, viz. configuration, is accomplished by running the configuration script. This script sets the required paths and generates a *makefile* for each module using a template *makefile* and an options file.

The second step, viz. building, is accomplished by running the build script to create the necessary libraries and the executables. The configuration and building is done separately for libraries, source modules and utility modules, however, the process followed for configuration and building is the same for all the three components.

To simplify the installation process of CFSv2, a single installation script (installer) to perform the two-step installation of all the components is provided with the CFSv2 suite. The step-wise installation procedures, quick as well as detailed, along with debugging tips are given below:

4.2.2.1 Quick Installation

1. Unzip and untar the CFSv2 zipped tar file and go to build directory present in CFSv2.

   $ cd $CFSROOT/build CFSROOT is the top directory of CFSv2 software.

2. In order to change the compiler and/or its options, open the corresponding options file and make the necessary changes. For example, to change the compiler/compiler options of libraries, open the file “options-lib” and make the necessary changes.
3. Prior to the installation, keep note of the full paths of CFSROOT, NetCDF include and library directories and MKL directory.
4. Install CFSv2 software system by running the installer.

   ```
   $ ./install.sh
   ```

5. After the successful installation, user can run CFSv2 using the run instructions described in section 4.3.

4.2.2.2 Detailed Installation

1. Get the CFSv2 zipped tar file. Unzip and untar this file to your system. We call the directory, where the model is untarred, as 'CFSROOT'.
2. Check whether the system requirements, mentioned in section 4.2.1, are met by the target machine.
3. Descend to the root directory of the CFSv2 software tree and then go to the directory “build”.

```
$ cd  $CFSROOT/build
```

The build directory contains installation script and three options files for model libraries, source modules and the utility modules.

4. In order to change the compiler and/or its options, open the corresponding options file and make the necessary changes. Make sure the changes made are consistent for all model components; else it may lead to run-time/incompatibility errors.
5. Prior to the installation, keep a note of full paths of CFSROOT, NetCDF include and library directories and MKL directory.
6. Install CFSv2 software system by running the installer.

```
$ ./install.sh
```

a) The installer starts by asking the user to enter the full paths of CFSROOT, NetCDF include and library directories and MKL directory.

b) Next, the user is asked to select the installation choice out of the following installation choices. **User can install each component separately or the entire model in a single step.** For the first time, the user may select option 4 for complete model installation. Other options may be selected in subsequent installations as per the requirement.

1: Install Libraries	-	To install all the libraries
2: Install Source modules	-	To install all the source modules
3: Install Utility modules	-	To install all the utility modules
4: Install All (Entire Model)	-	To install the entire model

A. Install Libraries, Source and Utility modules separately (If required)

How to build only libraries?

i. Enter '1' as your choice.

ii. Instead of building all the libraries user can also build a set of libraries, if required.

iii. The installer asks the choice from the user to build all the libraries, or one or more particular libraries.

```
Select the library which you want to install [1 or 2]
1: All
2: Select from list
```

iv. Press '1' to build all the libraries. Press '2' to build a set of libraries. If you press '2' then you must give a list of libraries which need to be built. Then, follow the instructions given by the installer.

v. After the successful installation of libraries, the libraries will be placed in the directory *"$CFSROOT/lib"* and the module files of the libraries will be placed in *"$CFSROOT/incmod".*

How to build only source modules?

i. Before starting the installation of source modules, user should make sure that all the dependency libraries are built successfully.

ii. Enter '2' as your choice.

iii. The installer asks the choice from the user to build all the modules, or one or more particular modules from the list.

```
Select the source module which you want to install [1 or 2]
1: All
2: Select from the list
```

iv. Press '1' to build all the source modules. Press '2' to build a set of source modules. If you press '2' then you must give a list of source modules which need to be built.

v. After the successful installation, all the executables will be kept in the directory “$CFSROOT/exec”.

How to build only utility modules?

i. Before starting the installation of utility modules user should make sure that all the dependency libraries are built successfully.

ii. Enter '3' as your choice.

iii. The installer asks the choice from the user.

```
Select the utility module which you want to install [1 or 2]
1: All
2: Select from the list
```

iv. After the successful installation, all the utility executables will be kept in the directory “$CFSROOT/util/exec”.

B. Install the entire model

i. Enter '4' as your choice. This option installs the entire model.

ii. The installer keeps updating the user about the progress of the installation and also reports about the success or failure of each library, source or utility module. The user can also check the installation log “log.install” in the directory “$CFSROOT/build” to verify whether the model is installed without errors or not.

iii. After the successful installation:

- ✓ The libraries will be placed in the directory “$CFSROOT/lib” and the module files for the libraries in “$CFSROOT/incmod”.
- ✓ All the executables of the source modules will be kept in the directory “$CFSROOT/exec”.
- ✓ All the utility executables will be placed in “$CFSROOT/util/exec”.

7. After all the libraries, source modules and utility modules have been successfully built, user can run CFSv2 using the run instructions described in section 4.3.

Note: If you choose to install the full model, the installer first installs all the libraries. After all the libraries are successfully installed, it installs the source modules. And at last, it installs the utility modules. If it fails to install any of the libraries, it continues the installation of rest of the libraries, however, it aborts without installing source modules and utility modules. If the installation of any of the source modules fails, then it continues with the rest of the source modules and aborts without installing utility modules.

The table below lists the source directories of the model parts and the respective executable files created by each part after successful compilation.

Table 4.2: Model source directories and executable files

S. No.	Directory Name	Executable Created
sorc/		
1.	*cfs_ao_coupler.fd*	*cfs_mlc_coupler*
2.	*cfs_atmos_fcst.fd*	*cfs_cdas_atmos_fcst*
3.	*cfs_mppnccombine.cd*	*cfs_mppnccombine*
4.	*cfs_ocean_mom4ice.fd*	*cfs_ocean_mom4ice*
5.	*cfs_overparm_grib.fd*	*cfs_overparm_grib*
6.	*cfs_psichi.fd*	*cfs_genpsiandchi*
7.	*cfs_separ3.fd*	*cfs_separ3*
8.	*cfs_tripo2reg.fd*	*cfs_tripo2reg*
9.	*global_chgres.fd*	*global_chgres*
10.	*global_sighdr.fd*	*global_sighdr*
11.	*ncep_post_new.fd*	*ncep_post*
util/sorc		
12.	*anomgb.fd*	*anomgb*
13.	*copygb.fd*	*copygb*
14.	*grbindex.fd*	*grbindex*

S. No.	Directory Name	Executable Created
15.	*wgrib.cd*	*wgrib*
16.	*ndate.fd*	*ndate*
17.	*nhour.fd*	*nhour*

4.2.2.3 Debugging Instructions

The following instructions will help you find out the possible cause of failure of full or part of the installation.

1. Open the file “log.install”, which is created in build directory as a part of the installation.
2. Locate the failed component's section in “log.install” file and find the error.
3. After fixing the error, you need not install the full model from the beginning again, instead you can install only the failed component(s).

4.3 Running the model

The CFSv2 model is programmed to run both in the coupled as well as atmosphere-only standalone mode. At CDAC, the model has been run in both coupled and atmosphere-only mode on the IBM-V (AIX) platform. On Linux platform, the model has been run in coupled mode.

4.3.1 Coupled Mode

As there is no GUI for the model, it needs to be run via command line. Before starting the model run, perform the following steps to setup the model run.

Step 1:

Place the initial conditions appropriately.

1. The atmospheric initial conditions viz. sigma and surface files, to be placed in $CFSROOT/com/ directory. Files accepted by the model are *siganl.gdas.YYYYMMDDHH* for sigma file and *sfcanl.gdas.YYYYMMDDHH* for surface file.

2. The ocean initial conditions to be placed in $CFSROOT/com/ directory as a tar file. File accepted by the model is *ocnanl.gdas.YYYYMMDDHH.tar*.
3. The time-dependent and angle-dependent satellite bias correction input files also need to be placed in the $CFSROOT/com/ directory. These files are named *biascr.gdas.YYYYMMDDHH* and *satang.gdas.YYYYMMDDHH* respectively.

Step 2:

Next, verify the values set in section 1 in the script "submit.sh" located in the $CFSROOT/exp directory. The *HOMEDIR* is set to the path provided during configuration. The configuration file is specified by the *CONFIG* variable. If running with another set of input conditions, *cdate* should be changed accordingly.

Step 3:

The next steps towards running the model program are:

a) Setting the length of the forecast and
b) The distribution of processors to run the three model components viz. Atmosphere, Ocean and Coupler.

These can be done in section 1 in configuration file *para_config_cfs* located at $CFSROOT/exp. This section contains the following variables:

- *run_length* : Length of the run (in hours)
- *n_procs* : Total number of processors
- *n_procs_o* : Number of processors for ocean
- *out_freq* : Output frequency (in hours)

a) The forecast-length of the model is described in terms of hours of forecast. The model takes this input from the variable *run_length*. Set this variable to appropriate value to specify the length of forecast.
b) To change the number of processors to run the individual model components perform the following steps:
    - Set variable *n_procs* equal to the total number of processors intended for model run.

- Set the number of processors for the Ocean model in variable *n_procs_o*. The number of processors for Coupler (generally 1) and Atmospheric component is computed by the script itself. For example: For the runs made at CDAC for total processors (*n_procs*) as 80, *n_procs_o* was set to 31. Hence, number of processors for atmosphere becomes 48.
- Lastly, change the environment variable corresponding to the processors in the PBS script (If you are using PBS on your cluster) namely $CFSROOT/exp/pbs.submit as

  ```
  #PBS -l nodes=10:ppn=8
  ```

Step 4:

Finally, the model run can be submitted to the PBS job scheduler by the PBS script located at $CFSROOT/exp/pbs.submit by issuing the following command:

```
$qsub pbs.submit
```

The model output and error log can be checked in the cfsv2.o$jobid and cfsv2.e$jobid files respectively, while the model part being run can be checked in $CFSROOT/exp/prcfsv2.runlog.

Intermediate Output logs: In directory com/YYYYMMDDHHgdasfcst1/

out → out.mpiexec.<FH_CYCL>

error → err.mpiexec.<FH_CYCL>

4.4 Model Output

The default configuration of the model outputs:

1. **Isobaric level data** (winds, temperatures, heights, etc.) at 37 pressure levels in the atmosphere on a 1° x 1° grid.
2. **Surface data** (2-m temperature, precipitation, snow, 10-m wind, surface fluxes, cloud cover, etc.) on 384 x 190 Gaussian (T126) grid.
3. **Ocean data** (temperature, salinity, currents) at 40 different depths and sea level height on 1° x 1° and 0.5° x 0.5° grid.
4. **Isentropic level data** (temperature, wind, humidity, etc.) at 16 isentropic levels on a 1° x 1° grid.

The model output directory is **$CFSROOT/com/**and the corresponding files are:

Sr. No.	**File**	**File naming convention**	**File type**
		Model Output	
1.	flx	flx<R><HH>.gdas.<IC-DATE>	GRIB
2.	sigf	sigf<HH>.gdas.<IC-DATE>	Binary
3.	sfcf	sfcf<HH>.gdas.<IC-DATE>	Binary
4.	ice	ice_YYYY_MM_DD_HH.gdas.<IC-DATE>.nc	NetCDF
5.	ocn	ocn_YYYY_MM_DD_HH.gdas.<IC-DATE>.nc	NetCDF
		Post Output	
6.	pgb	pgb<R><HH>.gdas.<IC-DATE>	GRIB
7.	ocn	ocn<R><HH>.gdas.<IC-DATE>	GRIB
8.	ipv	ipv<R><HH>.gdas.<IC-DATE>	GRIB

where,

R = one character signifying the resolution of the output file, which can be any of the following

f = model resolution (Gaussian Grid)

l = low resolution (1° x 1°)

h = high resolution (0.5° x 0.5°)

HH = two digit forecast hour

IC-DATE = Initial condition date in the format YYYYMMDDHH

4.5 Verification Data

CFSv2 has been implemented on CDAC's high performance Linux system, GG-BLR. The system has been run for short-range and medium-range simulations. These verification datasets are also provided alongwith the portable CFS model software system.

References

S. Saha, S. Nadiga, C. Thiaw, J. Wang, W. Wang, Q. Zhang, H. M. van den Dool, H.-L. Pan, S. Moorthi, D. Behringer, D. Stokes, G. White, S. Lord, W. Ebisuzaki, P. Peng, P. Xie *"The NCEP Climate Forecast System",* Journal of Climate, 2006, Vol. 19, No. 15, pages 3483-3517.

Kanamitsu, Masao, Wesley Ebisuzaki, Jack Woollen, Shi-Keng Yang, J. J. Hnilo, M. Fiorino, G. L. Potter, 2002 *"NCEP–DOE AMIP-II Reanalysis (R-2)." Bull. Amer. Meteor. Soc.*, **83**, 1631–1643.

Behringer, D. W., 2007 *"The Global Ocean Data Assimilation System at NCEP"*, 11th Symposium on Integrated Observing and Assimilation Systems for Atmosphere, Oceans, and Land Surface, AMS 87th Annual Meeting, San Antonio, Texas, 12pp

Moorthi, S., H. L. Pan and P. Caplan, 2001 *"Changes to the 2001 NCEP operational MRF/AVN global analysis/forecast system."* NWS Technical Procedures Bulletin, 484, pp14.

Mlawer E. J., S. J. Taubman, P. D. Brown, M.J. Iacono and S.A. Clough, 1997 *"Radiative transfer for inhomogeneous atmosphere: RRTM, a validated correlated-K model for the longwave."* J. Geophys. Res., 102(D14), 16, 663-16, 6832.

Ek M.B., K.E. Mitchell, Y. Lin, E. Rogers, P. Grunmann, V. Koren, G. Gayno, and J.D. Tarplay, 2003 *"Implementation of the Noah land-use model advances in the NCEP operational mesoscale Eta model."* J. Geophys. Res., 108, 8851, doi:10.1029/2002JD003296.

Wu, X., K. S. Moorthi, K. Okomoto, and H. L. Pan, 2005 *"Sea ice impacts on GFS forecasts at high latitudes."* Proceedings of the 85th AMS Annual Meeting, 8th Conference on Polar Meteorology and Oceanography, San Diego, CA.

Alpert, J.C., 2004 *"Subgrid-scale Mountain blocking at NCEP."* Proc. 20th Conf. on Weather and Forecasting, Seattle, WA.

Lott, F. and M. J. Miller, 1997 *"A new subgrid-scale orographic drag parameterization: its performance and testing."* Quart. J. Roy. Meteor. Soc., 123, 101-127.

Collins N., G. Theurich, C. DeLuca, M. Suarez, A. Traynaov, V. Balaji, P. Li, W. Yang, C. Hill, and A. da Silva, 2005 *"Design and implementation of components of Earth System modeling Framework."* The International Journal of High Performance Computing Applications, 19, #3, Summer 2005, pp-355-356.

McCormack, J.P., S.D., Eckermann, D.E. Siskind and T. McGee, 2006 *"CHEM2D-OPP: A new linearized gas phase photochemistry parameterization for high altitude NWP and climate models."* Atmos. Chem. Phys., 6, 4943-4972

Saha, Suranjana, and Coauthors, 2010 *"The NCEP Climate Forecast System Reanalysis."* Bull. Amer. Meteor. Soc., **91**, 1015–1057.

Murray, R. J., 1996 *"Explicit generation of orthogonal grids for ocean models."* Journal of Computational Physics, 126, 251–273.

Large, W. G., J. C. McWilliams and S. C. Doney, 1994 *"Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization."* Reviews of Geophysics, 32, 363–403.

Gent, P. R., and J. C. McWilliams, 1990 *"Isopycnal mixing in ocean circulation models."* Journal of Physical Oceanography, 20, 150–155.

Griffies, S. M., A. Gnanadesikan, R. C. Pacanowski, V. Larichev, J. K. Dukowicz, and R. D. Smith, 1998 *"Isoneutral diffusion in a z-coordinate ocean model."* Journal of Physical Oceanography, 28, 805–830.

Griffies, S. M., and R. W. Hallberg, 2000 *"Biharmonic friction with a Smagorinsky viscosity for use in large-scale eddy-permitting ocean models."* Monthly Weather Review, 128, 2935–2946

Hunke, E. C., J. K. Dukowicz, 1997 *"An Elastic–Viscous–Plastic Model for Sea Ice Dynamics." J. Phys. Oceanogr.*, **27**, 1849–1867.

Winton, M., 2000 *"A reformulated three-layer sea ice model."* J. Atmos. Oceanic Technol., 17, 525–531.

Mitchell, K. et al., 2005 *"NCEP implements major upgrade to its medium-range global forecast system, including land-surface component."* GEWEX newsletter, May 2005

Saha S et al. 2013 *"The NCEP climate forecast system version 2."* J. climate. Available from http://cfs.ncep.noaa.gov/cfsv2.info/CFSv2_paper.pdf.

Janakiraman, S., M. Ved, R. N. Laveti, P. Yadav and S. Gadgil 2011 *“Prediction of the Indian summer monsoon rainfall using a state-of-the art coupled ocean–atmosphere model.”* Current Science, 100, 354–362.

Catherine Thiaw and Suranjana Saha 2006 *“CFS Retrospective Forecasts: Time Series of Daily Data”* in the EMC/NCEP CFS public server www.cfs.ncep.noaa.gov/daily_cfs_data.doc

Catherine Thiaw and Suranjana Saha, 2007 *“CFS Retrospective Forecasts: Time Series of Monthly Means Data”* in the EMC/NCEP CFS public server” www.cfs.ncep.noaa.gov/monthly_cfs_data.doc

The Global Forecast Model (GFS) – Global Spectral Model (GSM): EMC/NCEP GFS Public server http://www.emc.ncep.noaa.gov/GFS/doc.php

Network Common Data Form library -
http://www.unidata.ucar.edu/software/netcdf/docs/faq.html#whatisit

Earth System Modeling Framework (ESMF) -
http://www.earthsystemmodeling.org/about_us/index.shtml