NMRF/TR/01/2017

TECHNICAL REPORT

Implementation and verification of Single Column Model at NCMRWF

A. Sandeep, E. N. Rajagopal and Aditi

January 2017

National Centre for Medium Range Weather Forecasting
Ministry of Earth Sciences, Government of India
A-50, Sector-62, NOIDA-201309, INDIA

Implementation and Verification of Single Column Model at NCMRWF

A. Sandeep, E. N. Rajagopal and Aditi

January 2017

National Centre for Medium Range Weather Forecasting

Ministry of Earth Sciences

A-50, Sector 62, NOIDA-201309, INDIA

Ministry of Earth Sciences
National Centre for Medium Range Weather Forecasting
Document Control Data Sheet

1	Name of the Institute	National Centre for Medium Range Weather Forecasting
2	Document Number	NMRF/TR/01/2017
3	Date of publication	January 2017
4	Title of the document	Implementation and Verification of Single Column Model at NCMRWF
5	Type of Document	Technical Report
6	No. of pages & figures, tables	32, 3, 3
7	Number of References	5
8	Author (S)	A. Sandeep, E. N. Rajagopal and Aditi
9	Originating Unit	National Centre for Medium Range Weather Forecasting (NCMRWF), A-50, Sector-62, NOIDA- 201309, India
10	Abstract	Single Column Model (SCM) is a useful test bed for investigating the parameterization schemes of Numerical Weather Prediction (NWP) models. The SCM is built by extracting the physical subroutines from the National Centre for Medium Range Weather Forecasting (NCMRWF) Unified Model (NCUM). SCM applications have usually been limited to single-grid nature where high-quality data is necessary to derive the essential boundary condition or forcing data. The expediency of SCM simulations is limited by the accuracy of prescribed large-scale observations. The global NCUM at N48L70 resolution is used to force the SCM and the simulations are carried out at an interval of 15 min for a forecast length of three days. The forecast of surface meteorological variables such as temperature (T), relative humidity (RH), wind speed (WS) and wind direction (WD) are obtained from SCM at Kanpur, India during February-April 2016. These forecasts are verified against the Automatic Weather Station observations and the forecast of variables available from the global NCUM. The SCM is able to predict the diurnal variation of the forecast variables (T, RH, WS and WD) however, the amplitude of variables are slightly overestimated relative to AWS observations. A very good correspondence is seen between SCM forecast variables against AWS observations and global NCUM forecast. The correlation coefficient (R) between SCM forecast and AWS observations for T, RH, WS and WD is 0.88, 0.93, 0.66 and 0.64, whereas R between SCM forecast and NCUM forecast for T, WS and WD is 0.91, 0.60 and 0.40, respectively. The present study highlights parameterization deficiencies and provides a useful tool for testing new or improved parameterizations.
11	Security classification	Unrestricted
12	Distribution	General
13	Key Words	Single Column Model, Automatic Weather Station, Verification of SCM forecast

Table of Contents

Abstract

Single Column Model (SCM) is a useful test bed for investigating the parameterization schemes of Numerical Weather Prediction (NWP) models. The SCM is built by extracting the physical subroutines from the National Centre for Medium Range Weather Forecasting (NCMRWF) Unified Model (NCUM). SCM applications have usually been limited to single-grid nature where high-quality data is necessary to derive the essential boundary condition or forcing data. The expediency of SCM simulations is limited by the accuracy of prescribed large-scale observations. The global NCUM at N48L70 resolution is used to force the SCM and the simulations are carried out at an interval of 15 min for a forecast length of three days. The forecast of surface meteorological variables such as temperature (T), relative humidity (RH), wind speed (WS) and wind direction (WD) are obtained from SCM at Kanpur, India during February-April 2016. These forecasts are verified against the Automatic Weather Station observations and the forecast of variables available from the global NCUM. The SCM is able to predict the diurnal variation of the forecast variables (T, RH, WS and WD) however, the amplitude of variables are slightly overestimated relative to AWS observations. A very good correspondence is seen between SCM forecast variables against AWS observations and global NCUM forecast. The correlation coefficient (R) between SCM forecast and AWS observations for T, RH, WS and WD is 0.88, 0.93, 0.66 and 0.64, whereas R between SCM forecast and NCUM forecast for T, WS and WD is 0.91, 0.60 and 0.40, respectively. The present study highlights parameterization deficiencies and provides a useful tool for testing new or improved parameterizations.

1. Introduction to Single Column Model

A Single Column Model (SCM) represents a single atmospheric column at one particular grid-point in a Numerical Weather Prediction (NWP) model. One of the major problems of NWP models lies in their scarce horizontal and vertical resolutions which are not suitable to predict the micro-scale phenomenon. The models are usually devoted to rejecting observations that are not representative of a relatively extended area. The initial profiles of mass and wind are forced during the assimilation process in order to fulfill some prescribed which is not realistic sometimes. Especially over inhomogeneous terrain, the state of the atmosphere at a specific location can be considerably different to that in the closest model grid point. One way to overcome these problems is the use of single column model (1-D), where the terms depending on the horizontal and vertical structure of the atmosphere are estimated from the outputs of operational 3-D models. SCM's are useful test beds to investigate the sensitivity of different parameterization schemes available for various NWP models; however, it can be useful to keep the large-scale atmospheric circulation fixed (Davies et al. 2013). A better assessment can be done through the impact of the local climate without impediment of large-scale feedback while testing the new parameterization scheme. On the other hand, SCM's are very inexpensive and can run on a powerful Personal Computer (PC) or an ordinary workstation. They go like the wind on a Cray (Randall 1994). SCM applications are usually limited to regions where high-quality observations are available to derive the necessary boundary condition or forcing data. In addition, the expediency of SCM simulations is limited by the accuracy of best estimate large-scale observations prescribed.

National Centre for Medium Range Weather Forecasting (NCMRWF) Unified Model (NCUM) can be considered to be a collection of many single-column models arranged to cover the entire globe and interacting with each other through a set of rules known as "large-scale dynamics". The global analysis fields available from NCUM are used to force the SCM which allowing the surface to be forced with time varying atmospheric conditions. When an SCM is forced with the NCUM global analysis the observed errors are either due to the chosen column physics or due to problems with the observations that are used as input. Despite this, relatively little attention has been paid to the critical evaluation and documentation of results from SCM derived from the operational NCUM model. Therefore, the aim of this technical report is to evaluate the performance of SCM in predicting the meteorological variables at a particular location. Also an inter-comparison of the diurnal cycle of different variables such as temperature (T), relative humidity (RH), wind speed (WS) and wind direction (WD) predicted by SCM is performed against AWS observations. Further, these variables are also predicted using the global NCUM at the same location and are compared with that predicted by SCM at that location.

The technical report gives the answers to the following questions: 1) How to generate an appropriate namelist file which is prerequisite to run SCM; 2) How to compile and run the SCM at different stations; 3) Verification of SCM forecast against AWS observations and global model forecast variables. Special emphasis is given to the JULES parameters, vertical levels, time resolution and post-processing of SCM output. In the material that follows, Section 2 gives a brief description about the treatment of different forcings. The main features of SCM include how to compile and run SCM and information about namelist file used to run it are given in Sections 3-6. Finally the verification of SCM forecast against a forecast of global NCUM and AWS observations is given in Section 7.

2. Treatment of different forcings

The SCM can be forced by various forcings viz., observational forcing (large-scale forcing), statistical forcing and Geo-strophic forcing (Wong 2010). The uncertainties inherent in these forcing data products have an unknown and possibly significant effect on SCM runs. The large-scale forcing can be determined by using vertical profiles of different variables estimated from an observational/analysis data set. The present study utilizes vertical profiles of potential temperature (θ), specific humidity (q), 3-dimensional components of wind (u, v, w) and there tendencies temperature (t_inc), specific humidity (q_star), 3-d winds (u_inc, v_inc, w_inc) obtained from NCUM global analyses. These are supplied through a namelist file known as namelist.scm at each time step. In addition to this, initial profiles of pressure (p) and vertical levels of potential temperature (θ) and density (ρ) are also supplied at the first time step. The large-scale forcing (in terms of profile) and tendencies are calculated at regular time intervals (6 h) for each variable at user specified grid point through the bi-linear interpolation technique. Note that this time interval (obs_pd) is not necessarily the same as the SCM model time step. Along with the large-scale forcing the surface forcing can also be provided as time series of either surface temperature (T), sensible heat flux (H) and latent heat flux (E). The present setup of SCM uses only the large-scale forcing (Randall and Cripe 1999) and surface forcing at 6 h temporal resolution. A hybrid height vertical co-ordinate system is chosen with terrain following at lower level and flattening out near the top of the atmosphere. The Charney-Philips grid is used so that the variables u, v and p are obtained on the ρ levels (model levels) and variables (θ, q and w) are obtained on the θ levels (model levels + 1). A total 70 model vertical levels are used and the choice of the vertical resolution is non-uniform (see Figure 11; Rajagopal et al. 2012). Changes in the model levels can be done through a base file similar to the global model in which we need to supply z_top_of_model (m), first_constant_rho_level (first level at which levels are a constant height across grid points in a global model), θ and ρ levels.

The initial profiles of T and q are first input and vertical profiles of winds are calculated at each level by using the Geo-strophic approximation equations (Wong 2010) at each time step. Moreover, initial values of wind are normally set through zonal Geo-strophic wind (u_g) and meridional Geo-strophic wind (v_g) values if the geoinit logical is set "true" in namelist.scm. In the current SCM set up it is set as true, so the variables u_g and v_g are fixed as 5 m s^{-1}. No statistical forcing is used in the SCM run. However, it is possible to run SCM with more than one type of forcing at once. For example a combination of observational and Geo-strophic forcing is necessary for boundary layer experiments. Note SCM will warn if you are running with multiple forcings because some combinations may not give sensible results. Depending on the case studied advective tendencies both in horizontal and vertical, radiative tendencies or surface boundary conditions (e.g. Surface fluxes or surface temperature) may be prescribed additionally.

3. Main features of the SCM

The Unified Model User Interface (UMUI) is used to run the global NCUM where it includes several steps. A flow chart showing the main calling structure is shown in Figure 1 (source: Wong 2010). The routine SCM_SHELL is the top level routine which reads various parameters from the UMUI before calling the SCM_MAIN. In SCM_MAIN the physics routines are called twice: 1) to initialize various variables before the time stepping; 2) during the time stepping. All the subroutines such as physics routines, forcing routines and initialization routines are called within this program. The key physics routines are MICROPHYSICS_CTL (cloud and large-scale rain schemes), NL_RAD_CTL (long-wave and short-wave radiation), NL_CONV_CTL (convection scheme) and NL_BL_CTL (boundary layer scheme). Further details about the subroutines and control files are described in the sub-sequent sections.

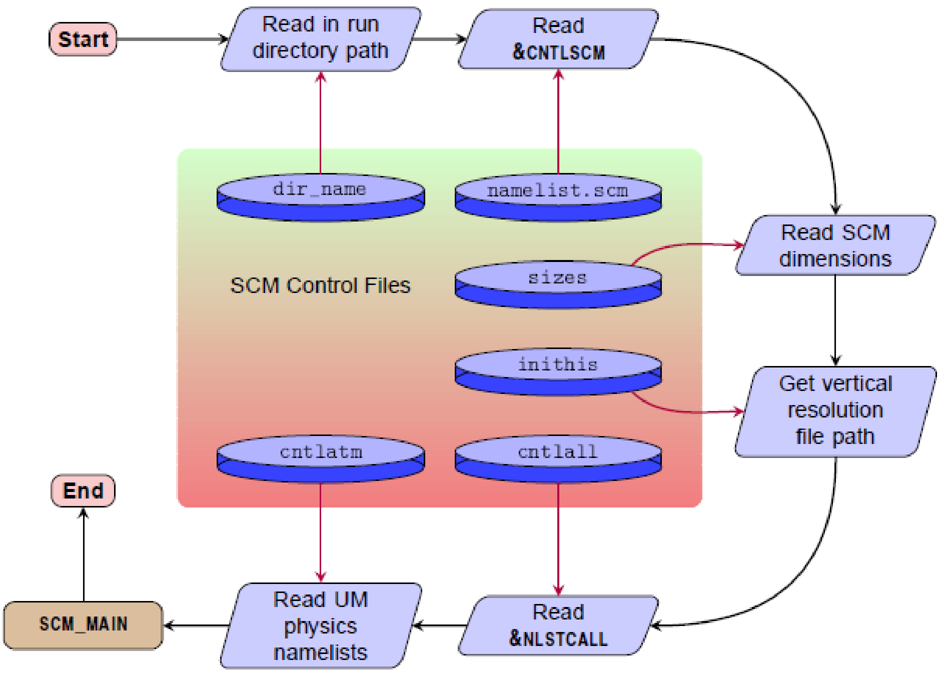

Figure 1: Flowchart for top level flow routine SCM_SHELL.

4. Compiling the SCM

The first and foremost step to run the SCM forecast is the successful compilation of SCM. It can be done through UMUI in which few variables/paths can be modified/set accordingly. It is reiterated that the UMUI currently only provides support in building the SCM executable. Further, it generates necessary control files to run the SCM forecast at different grid points. Most of the elements required to use the SCM are transparently provided by the UMUI. Some of these elements (ancillary data and dumps) are not yet included. The key changes required to compile the SCM are tabulated in Table 1. An interesting feature of SCM is that, once SCM compiled successfully which can be used at any grid point in the global domain. The SCM compilation is more reliable which includes the entire code into it. The present UMUI provides support in building the SCM executable only which is further used to run the SCM at user location with the supplementation of namelist.scm. In the present study N48L70 job of NCUM vn 8.6 is used to compile the SCM.

Table 1: UMUI Job modifications

		Control Parameter	Changes to be made
A) User information and submission method			
	a) General details		
		Target Machine user-id:	username
		Mail-id for notification of end-of-run	username@ncmrwf.gov.in
	b) Job Submission method		
		Define submission method	LSF
		Host name	ncmlogin3
		Target machine name	ibm
		$UM_MACHINE	ibm-ifort-ncm
B) Input/Output Control and Resources			
	a) Time Convention and SCRIPT Environment Variables		
		DATAM	/gpfs2/home/username/UM/vn8.6/output/$RUNID
		DATAW	/gpfs2/home/username/UM/vn8.6/output/$RUNID
		UM_TMPDIR	/gpfs2/home/username/tmp/um/vn8.6/temp_$RUNID
		MY_OUTPUT	/gpfs2/home/username/um_run_log
C) FCM Configuration			
	a) FCM extract directories and Output levels		
		UM_SVN_URL	fcm:um_br/dev/moum/r2990_ncm_BHPC_UM8.6/
		UM_SVN_BIND	$UM_SVN_URL/src/configs/bindings
		UM_CONTAINER	$UM_SVN_BIND/container.cfg
		UM_OUTDIR	$/gpfs2/home/username/tmp/UM8.6/UMUI_extracts
		UM_ROUTDIR	/gpfs2/home/username/UM/vn8.6/UM_runs
		FCM Extract output will be sent to	$UM_OUTDIR/ext.out
D) Atmosphere			
	a) Model Resolution and Domain		
		Horizontal	
		Select Area Option	Single Column Model
		Vertical	
		Number of levels	70
		Number of wet levels	70
		Number of ozone levels	35
		Type of vertical interpolation	Linear with no extrapolation
		Number of boundary layer levels	50
		Number of non-local boundary layer levels	30
		Number of deep soil levels	4

(excluding surface)	
Number of cloud levels used in radiation	70
Vertical levels Options	User defined set
Directory	/gpfs1/home/moum/vn8.6/ctldata/vert
File	vertlevs_UKV_L70
Soil levels	1) 0.10 2) 0.25 3) 0.60 4) 2.00
Single Column Settings	
Specify path for forcing namelists	/gpfs2/home/username/SCM/template.scm
Output Data Files	NCUM_SCMoutput.dat
b) Scientific Parameters and Section	
Section by Section Choices	
Section 1: SW Radiation	
Options for multiple calls to radiation	Time stepping scheme
Number of segments	20
Number of bands	6
Number of times per day to calculate increments (Prognostic)	8
Number of times per day to calculate increments (Diagnostic)	24
Minimum albedo of sea ice	0.60
Maximum albedo of sea ice	0.80
Temperature range over which the albedo varies linearly between max and min values	10.00
Orographic Correction	Flat surface
Gen2: SW Radiation	
Solar constant	1365.00
Directory	${UM_RAD_SPECTRAL_GL}/ga3_1
Prognostic file	sp_sw_ga3_0
Diagnostic file	sp_sw_cloud3_0
Specify mass-mixing ratio of oxygen	0.2314
Section 2: LW Radiation	
Options for multiple calls to radiation	Time stepping scheme
Number of segments	20
Number of bands	6

		Number of times per day to calculate increments (Prognostic)	8
		Number of times per day to calculate increments (Diagnostic)	24
		Gen2: LW Radiation	
		Directory	${UM_RAD_SPECTRAL_GL}/ga3_1
		Prognostic file	sp_sw_ga3_0
		Diagnostic file	sp_sw_cloud3_0
		Section 3: Boundary Layer and Surface Processes	
		Choose version	9C
		Select type of SBL mixing scheme	RiSc

The SCM code is included with the NCUM code and user can modify the SCM routines using FCM branches set in the UMUI or modifying the working copy directly by hand. Once all the settings are included in the SCM setup, it creates a ".leave" file after running. This file provides information on the compilation should be processed in the output directory if it exists. When the job is compiled successfully the UMUI generates four SCM control files (SIZES, INITHIS, CNTLATM and CNTLALL) as shown in Figure 1. All these SCM control files contain too much information and hence require some modification. These control files with the modifications are given in Table 2. Some of other control files such as dir name and namelist.scm are provided by the user as they are not generated by UMUI (Section 6). The UMUI only provides support in creating the executable file and cannot set up all the information required by the SCM to run the model.

Table 2: SCM control files after UMUI compilation.

Control File	Changes
CNTLALL	No change required
CNTLATM	No change required
INITHIS	No change required
SIZES	&NLSIZES levels to be modified if different than specified in UMUI
SCM_SET	Add VERT_LEV path Add scm_nml path Add strm_filename

5. Genesis of namelist.scm

It is clear from above sections that the compilation of SCM generates only executable file, however, there is a need of initial conditions are to be supplied to run the SCM forecast. The necessary initial conditions are supplied through a separate file called “namelist.scm” generated by Genesis software. It creates the namelist file from NWP model fields and is widely used in other operational centres. This is a useful tool for development and testing of model parameterizations. It simplifies the preparation stages of namelist file generation and enables the user to create their own environments as a framework for SCM studies which are based on approximate real world (or model world) conditions. This software was developed on the Fortran platform which contains various sub-program files. The present study uses an upgraded version of Genesis to generate the namelist file. Input data are taken from required global fields (T, RH and 3-d wind components) in NetCDF format and set out on regular lat/lon grids (no staggering) and pressure levels (in Pascals). The genesis software requires the 4 namelist files which are pre-requisites to generate namelist.scm:

1. files.inp - lists the names of the input files in order: mslp, z, u, v, t, q
2. dates.dat - lists the dates corresponding to each time slice of model run period.
3. base.inp - is a namelist containing input specific to the model run including switches for the inclusion of particular forcing fields as required.
4. template.scm – is a duplicate of namelist.scm which consists all the required parameters to be included.

The makefile is set up to compile genesis 2.1 on BHASKARA high-performance computing system. Some modifications are done in the FC, LIBDIR and INCDIR paths according to local versions of f90 compiler and NetCDF libraries. Once all paths are set up, “make all or make genesis”: compiles all the objects and genesis executable with entitle “genesis”. This executable along with the above mentioned namelist files (files.inp, base.inp, dates.dat and template.scm) are pre-requisites to make the namelist.scm file. The syntax of genesis is given below

./genesis –d (debug mode) –t (no.of time steps) –l (no.of vertical levels) –x (no.of latitude points) –y (no.of longitude points) –X (user latitude) –Y (user longitude) –D (use date file) –R (convert relative humidity to specific humidity) –U (add namelists file for various parameters) < base.inp > out.log

An additional output files are created prior to the generation of the output namelist files called “genesis.scm” and “charney.scm”. The genesis.scm contains profiles of z, p, t, theta, q, u, v, u_g and v_g for each time slice for examination at the user specified latitude and longitude. The charney.scm contains profiles of p, z, T, θ, q, u, v and w at the last time step. In addition to this, a log file is generated which includes the flow of the genesis program in generating the namelist files. This tool is sensitive to vertical velocities, therefore it is recommended use a short time step if SCM runs of more than 3 days or so. For longer time steps the vertical advective component for T and q in interp_linear.f90 (horizontal tendencies are worked out) can be comment to get good simulations. At the end an output file namelist.scm is generated through Genesis 2.1. The major groups of namelist.scm are tabulated in Table 3. A detailed description about the each group of namelist.scm can be found in Wong (2010). The namelist.scm is not yet ready to run the SCM since it included only the large-scale forcing, surface forcing and Geo-strophic forcing. It also requires some other important parameters such as &INJULES parameters (canopy, catch, tstar_file, z0_tile, snow_tile), &INPROF parameters (sil_orog_land, ho2r2_orog, canopy_gbi, snodepi, tstari, z0m_seai, ice_fract and di), profiles of sub surface parameters (soil moisture, soil temperature, frac_type, lai and canht) and ozone profile which are retrieved from “.astart” file. From this file, all the above parameters are retrieved at the first time step and added to namelist. scm through shell scripts. Furthermore, the choice of surface is also included at the given grid point through SCM grid point configurations (Land-soil selected for the current SCM setup). If the grid point is not at sea, further details regarding land surface fraction at that particular grid point are also necessary. The parameter ichgf is the ratio of time between observation forcing profiles and to time step given in the “&RUNDATA” group. Note that if the user modified time step length and is necessary to change ichfg. To ensure the forcing data is appropriate for SCM run variable arrays in the namelist.scm are dimensioned with model_levels_nml and should be consistency with model_levels as specified in SIZES. Although some arrays are dimensioned as (row_length, rows) implying that the multiple SCM columns can be run. The “&CNTLSCM” group is read by routing SCM_SHELL so that some details are known prior to calling SCM_MAIN. Further, the SCM diagnostics system is controlled by &DIAGS group. If the user running without this group, the system will operate on defaults and an output file will be produced. The first variable in &DIAGS group is main_diag_switch is an integer if zero/nonzero means that the diagnostic system is off/on (diagnostics are neither processed nor output). By default it is non-zero and the diagnostic system is on.

Table 3: The variable arrays/groups of namelist.scm

S. No	Group	Parameters/Variable arrays
1	&CNTLSCM	nfor, model_levels_nml, l_ts_log, land_points
2	&INDATA	year_init, mon_init, day_init, min_init, sec_init, lat, long, soil_type, gridbox_area
3	&RUNDATA	exname_in, exname_out, ndayin, nminin, nsecin, timestep, ntrad1, ntml, zh, tstar_sea, tstar_land, fland_ctile, albsoil, orog, ozone
4	&LOGIC	ancyc, altdat, obs, obs_surf, prindump_step, prindump_day, land_sea_mask, land_ice_mask, soil_mask, geoforce, geoinit, grafdump_step, local_time, l_spec_z0
5	&INJULES	gs, canopy, rgrain, smi_opt, smcli, frac_typ, canht, catch, snow_tile, lai, z0_tile, infil_tile, tstar_tile
6	&INPROF	canopy_gbi, smci, t_deep_soili, sil_orog_land, tstari, ho2r2_orog, ice_fract, di, u_0, v_0, snodepi, nml_inprof_theta1, z_tom_nml, eta_th_nml, eta_rh_nml, ui, vi, wi, theta, qi, p_in
7	&INGEOFOR	ug_opt, vg_opt
8	&INOBSFOR	old_vertadv, l_vertadv, obs_pd, rlx_t, rlx_q, rlx_u, rlx_v, rlx_w, t_inc, q_star, u_inc, v_inc, w_inc
9	&DIAGS	strm_dumpstep(1) = 1, , L_SCMDiag_gen = .true. , L_SCMDiag_conv = .true. , L_SCMDiag_rad = .true. , L_SCMDiag_bl = .true. , L_SCMDiag_lsp = .true. , L_SCMDiag_lscld = .true. , L_SCMDiag_forc = .true. , L_SCMDiag_incs = .true. , L_SCMDiag_pc2 = .true. , L_SCMDiag_surf = .true. , L_SCMDiag_sea = .true. , L_SCMDiag_land = .true. , strm_format = 4 /

There are 12 logicals for turning on packages of diagnostics allowing selected subsets of the complete set of diagnostics to be output. These packages typically represent sets of diagnostics from different physics sections of the model or ones which are useful to be available together. Only "l_scmdiag_gen" is default true, switching on 50 of the approximately 324 diagnostics available as standard. Some diagnostics are only available when certain model switches are on, for example land diagnostics are only available when running over a land point. These logicals will be reset to false and a warning message displayed if they are switched on in the namelist.scm file but the model setup makes those diagnostics unavailable. The packages are not unique and many diagnostics are available from more than one package, for example increments from the convection scheme are available both from the increments package and the convection package. The integer array strm_dumpstep dictates the dumping period for each stream. This dictates the processing period for the diagnostics being sent to that stream.

6. Running SCM

It is clear from the previous sections that how to compile the SCM and generation of an appropriate namelist.scm file. In general, those variables which you may wish to change during the setup of a run are contained in namelist.scm. With the successful completion of these steps, we can run SCM job on BHASKARA High Performance Computer (HPC) system with the supporting control files. To run a SCM job, the executable requires access to the files listed below. Note that all these files are pre-requisites to run SCM forecast.

1) namelist.scm (Generated through genesis)
2) scmjob.lsf (Script to run the Job)
3) jobname.exe (Compiled through UMUI; SCM executable)
4) UMUI control files (CNTLALL, CNTLATM, SIZES, INITHIS and SCM_SET)

The SCM executable reads the file directory and the output file name and its path are given through SCM_SET. Once all above files are available, it is possible to run the SCM by invoking the executable. The script file scmjob.lsf is created to run the SCM at user specified location. It sets up the various environment variables which are required. The SCM gives the output data in .dat or NetCDF format which is set in the “&DIAGS” group in the namelist.scm. Two additional files (scumlist.dat and domain.dat) are also generated along with output data. The first contains information about the exact formatting of the data in the format-zero data files (including a complete list of all diagnostics with their names). The second contains information on the size of the model domain (row_length, rows), the vertical levels and the orography needed to calculate physical positions from model points. The output may also be split across a number of files in order to facilitate the separate post-processing of classes of diagnostics. NCUM storage handling and diagnostic system (STASH) is to be set up inside the job (Rajagopal et al. 2012). All the details about running the SCM through shell script at user latitude and longitude is described in Appendix A.

7. Validation of SCM output

It is clear from above sections the SCM runs are successfully developed in NCMRWF on trial-basis by extracting the physical sub-routines from the global NCUM model. SCM can run at any model grid point in the global domain. For example Kanpur (26.47°N latitude, 80.33°E longitude, 142 m above mean sea level) is an urban/industrial site located in the central part of the Indo Gagnetic Belt and SCM run at this location. Further, the SCM outputs are validated against Automatic Weather Station (AWS) observations and NCUM global-model forecast over Kanpur station during February-April 2016 first through a typical case study, later combining all clear-sky.

7.1. Intercomparison of SCM, NCUM and AWS outputs – a case study

SCM results are being tested in the clear-air conditions because dense clouds (and rain) complicate the testing. Therefore, three clear-sky days (19-22 February 2016) are selected based on the incoming solar radiation at the noon time (> 900 W m^{-2}) to verify the SCM output with the respective station data. The surface meteorological variables such as temperature (T), relative humidity (RH), wind speed (WS) and wind direction (WD) are extracted from SCM, AWS and NCUM outputs at Kanpur station. Note that RH data is not available at Kanpur for NCUM forecasts. Figure 2 represents the diurnal variation of surface variables extracted from SCM, AWS and NCUM outputs during 19-22 February 2016 over Kanpur. It is clear from the Figure 2, the SCM is able to predict the diurnal variation of all the forecast variables. As seen in Figure 2a, for both daytime and nighttime hours, the SCM forecast of T is slightly overestimated relative to AWS observations and NCUM forecast. In Figure 2b, with the advancement of forecast days, the difference of RH between SCM forecast and AWS observations is increased monotonically. The amplitude of diurnal cycle for an SCM forecast of T and RH is nearly same during 19-22 February 2016. As seen in Figure 2c, the SCM forecast of wind speed is slightly underestimated with respect to AWS observations, however, large differences are found between the SCM forecast of wind speed and NCUM wind speed forecast. Though the values of WS in NCUM forecast are two factors greater than SCM forecast, but the amplitude of WS is monotonically increasing with advancement of the day. As seen in Figure 2d, the diurnal variation of WD is quite distinct with noticeable differences between SCM, AWS and NCUM in both daytime and nighttime. During 19-22 February 2016 period, SCM forecast of WD is westerly-northwesterly, whereas AWS (NCUM) observations (forecast) show predominantly westerly (southerly) during the daytime and southerlies (easterlies) in nighttime, respectively. Given such a large variation in wind direction, there is need to be careful when interpreting the SCM forecast of WD. These wind differences can be inferred that, during the SCM run much more attention/care is necessary in the calculation of advection tendencies appropriately. Added to that, the temporal resolution (6 h) is also may not be sufficient to simulate the winds.

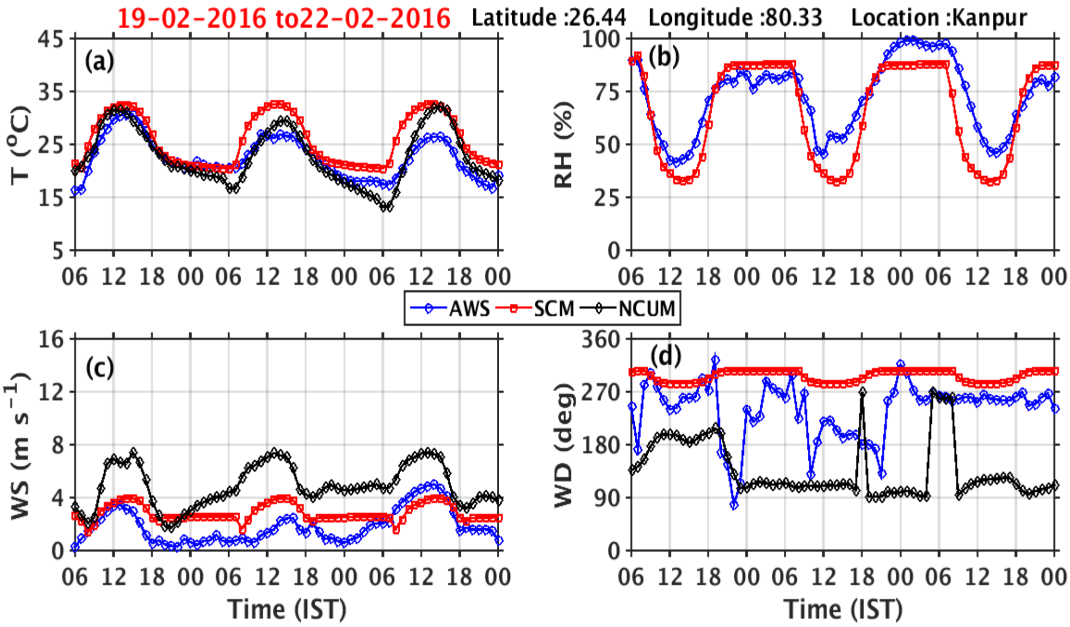

Figure 2: The diurnal variation of surface meteorological variables (T, RH, WS and WD) of SCM forecast (red), AWS observations (blue) and NCUM forecast (black) during 19 to 22 February 2016 period.

7.2. Intercomparison of SCM, NCUM and AWS outputs – Mean characteristics

The surface meteorological variables (T, RH, WS and WD) obtained from SCM forecast, AWS observations and global NCUM forecast are hourly averaged for selected clear sky days. During February-April 2016 a total 30 clear sky days are selected based on the incoming solar radiation and the availability of NCUM forecasts. A very good correspondence is seen between SCM forecast variables against AWS observations and global NCUM forecast. Figure 3 shows the scatter plot between the hourly averaged SCM forecast of surface meteorological variables (T, RH, WS and WD) against AWS observations (blue circles) and with NCUM forecast (red squares) over Kanpur. It is clear from Figure 3, the SCM forecast of T and RH showing very good agreement with AWS observations representing the results are more reliable. On the other hand, the SCM forecast of winds showing relatively lower agreement with AWS observations. The correlation coefficient (R) between SCM forecast and AWS observations for T, RH, WS and WD is 0.88, 0.93, 0.66 and 0.64, whereas, R between SCM forecast and NCUM forecast for T, WS and WD is 0.91, 0.60

and 0.40. The R is found to be higher (lower) for SCM forecast of RH (WS) with respect to AWS observations, respectively. Similarly the R is found to be higher (lower) for SCM forecast of T (WD) with respect to global NCUM forecast, respectively. Interestingly, all these correlations warrants further detailed studies by fixing some bugs in the namelist.scm, improving the time resolution, advective tendencies and spin-up time. Derived fields such as large-scale vertical motion and advective tendencies are particularly difficult to determine because they involve horizontal derivatives of the main flow. Aliasing can result from inadequate coverage in either space and time. These simulations are able to highlight parameterization deficiencies and provide a useful tool for testing new or improved parameterizations.

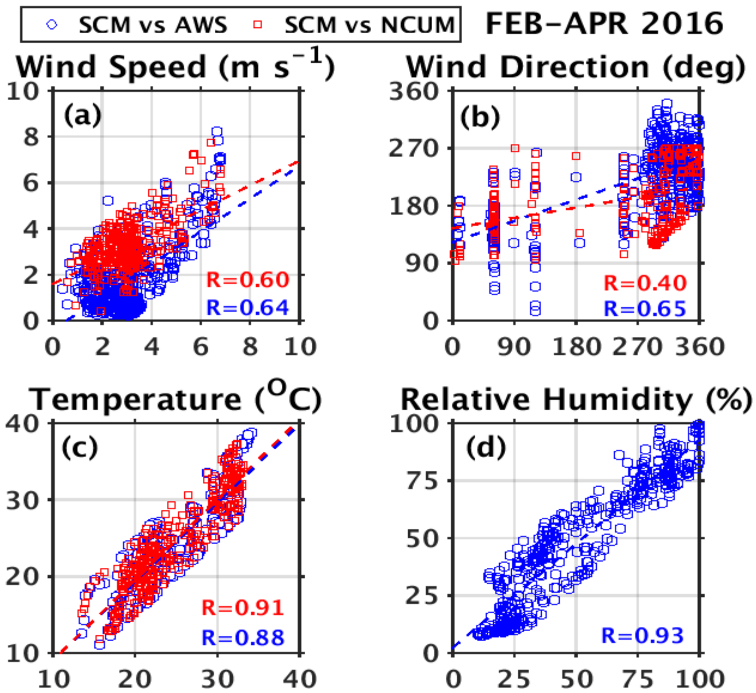

Figure 3: Scatter plot between hourly averaged SCM forecast of surface meteorological variables (R, RH, WS and WD) with respect to AWS observations (blue open circles) and global NCUM forecast (red open squares) during the study period over Kanpur. The bottom right side of each sub panel shows the value of correlation coefficient (R) between respective sets.

Acknowledgements:

The authors wish to thanks Dr. J. P. George, Dr. A. K. Mitra, Mr. G. R. Iyengar, Dr. Ricky Wong and Dr. Michael Whitall for their important suggestions and valuable comments for improvisation of SCM setup at NCMRWF. We extend our thanks to Dr. V. Barras, Bureau of Meteorology for developing Genesis 2.1 tool.

Appendix A: SCM run through shell script

```
######-------SCM NCMRWF--ASANDEEP
#!/bin/bash
START=$(date +%s)
## Shell script for extrating
u,v,t,q,z,mslp from um analysis ##
#Note1: Before running this script
you have to compile the SCM Job
through UMUI xbqla-UM8.6 N48L70...
#Note2: Once UMUI job submitted
successfully then a folder
avialable in
$HOME/umui_runs/jobname_jobid
(SCMDIR)
#Note3: copy
$HOME/UM/vn8.6/UM_runs/user/job/um
atmos/bin/job.exe into SCMDIR
#Note4: Check in SCMDIR:
scmjob.lsf, SCM_SET, CNTLATM
(check line by line in all three
programs)
#Note5: Check files in Rundir:
base_master.inp, template.scm
(This two files are mandatory to
create namelist.scm)
#Note6: The following script will
be take care of everything plot
the output.....
#Note7: Please check
globalpars.tcl (2 3 5 or 3 4 6)--
equilvalent to u,v,temp (daily 1hr
global forecast data)
#Note8: Check the template.scm is
with advection or without
advection..
#Note9: Check advday file exist or
not
set -x
echo "Give the year, mon, day, lat
lon, loc only,....dont change
others....formats are in brackets"
echo "Initial setup should be give
by usere accordingly (USER
INTEREST):"
echo "Check the Initial month (mon
and mont, year, location,
latitude, longitude..."
## Input details supplied to the
SCM to create a namelist.scm
year=2016 ##Give the year of the
experiment (Year format yyyy)
mon=02    ##Give the month of the
experiment (Month format mm)
mont='feb' ##Give the first three
letters of month in lowercase(mmm)
##tht=240   ##Give the terrain
height of a site location (m)
export loc=Kanpur ## Specify the
name of the site (n no.of
characters)
export lat=26.43  ##Give the
latitude of a site to be run
(xx.yy)
export lon=80.33  ##Give the
longitude of a site to be run
(xx.yy)

## Creating a directory to store
the output files accordingly
## Do this step only once in a
month (pre-check if already exist
remove the following line and run)
OPDIR=${loc}_${mont}${year}
#mkdir -p $OPDIR
echo "A new directory is created
as " $OPDIR

#Loop repetition for SCM on daily
basis
## Starting the analysis of SCM
forecast for each day..
for day in {1..28} ##Give on which
days you have to run the SCM
(start date..end date)
do
echo "THE BEGINNING"
echo "For more details contact
asandeep@ncmrwf.gov.in,
rajagopal@ncmrwf.gov.in"
echo "--------------------------
SCM SETUP - SANDEEP--------------"
echo "Clear the RUNDIR which are
not useful: Initialising the
setup"
rm -f f*.nc files.inp base.inp
dates.dat genesis.scm charney.scm
namelist.scm analaysisfiles_new.nc
rm -f rundata_xbffc.nc
jules_xbffc.nc inprof_xbffc.nc
levs_xbffc.nc xbffc.astart
namelistcorr.scm namelistfinal.scm
ozone.nc
rm -f NCUM_SCMoutput.nc
metpars_NCUM_SCMoutput.ctl
metpars_NCUM_SCMoutput.dat
radpars_NCUM_SCMoutput.dat
radpars_NCUM_SCMoutput.ctl
rm -f tht tht_xbffc.nc
metpars_NCUM_SCMoutput_corr.ctl
radpars_NCUM_SCMoutput_corr.ctl
metpars_NCUM_SCMoutput_corr.nc
radars_NCUM_SCMoutput_corr.nc
rm -f gu.nc gv.nc gtemp.nc
globalpars_umglaa_pe024.nc
globalpars_umglaa_pe024.ctl
```

```
globalpars_umglaa_pe024.dat
namelistcorrected.scm

echo "SETup the directoreis:"
echo "Check the paths accordingly
to user usuage:"
export RUNDIR=$PWD
export
SCMDIR=/gpfs2/home/asandeep/umui_r
uns/xbqla-071102419
export
UMDIR=/gpfs3/home/umfcst/NCUM/fcst
export TCLDIR=/gpfs2/home/asandeep
export
convsh=/gpfs1/home/moum/um_visual/
convshR81.91

echo "Running dir where the
original script is (Current
directory):" $PWD
echo "SCM directory where
compilation is done (Do one
time):" $SCMDIR
echo "UM directory where *.pp0,
and *.astart file contains: "
$UMDIR
echo "The current subset.tcl
directory " $TCLDIR

 if [ $day -lt 10 ]
 then
  dayy=0$day
 else
  dayy=$day
 fi

## -------------------------------
---- PART 0................Terrain
height retrieval
##### Extracting the terrain
height from the first astart file
echo "------------------ "
echo "PART 0 is running: "
echo "------------------ "
echo "Extracting the terrain
height wait"
tdate=$year$mon$dayy
cdate=00Z${dayy}${mont}${year}
echo "the terrain retreived date
is:" $tdate
du -sh
$UMDIR/$tdate/00/xbffc.astart |
cut -c1-2 > filesize
read xbffcsize < filesize
echo "Checking the file size of
xbffc.astart to start the program"
 if [ $xbffcsize -eq 0 ]
 then
  echo "xbffc.astart file is not
accessible for the day " $tdate
  echo "We are skipping this day"
 else
ln -fs
$UMDIR/$tdate/00/xbffc.astart .
sleep 5s
convshR81.91 terrainheight.tcl
xbffc.astart
----------------------------------
----------------------------------
#  Convsh script inputfile.tcl
#  Extract the parameters required
for the SCM from a xbffc.astart
#  file and retrieving according
to different sections and
converting
#  them into particular netcdf
file format and path where the
#  data is stored in directory
/gpfs4/home/asandeep/SCM/scmincomp
ass/
#  and write out as a netCDF file
# Similar statement everywhere
when calling convsh script
#  Read input file
   #readfile $filetype $infile1
#  Write out terrain height to a
netCDF file
   #writefile $outformat $outfile1
$fieldlist1
#  Remove input file information
from Convsh's memory
   clearall
----------------------------------
---------------------
#! /usr/bin/env convsh
set outformat netcdf
set filetype 0
#  RUNDATA@@ 25-orog;
set fieldlist1 "25"
foreach infile1 [glob
xbffc.astart] {
set outfile1 "tht_[file tail [file
rootname $infile1].nc]"
   readfile $filetype $infile1
   writefile $outformat $outfile1
$fieldlist1
   clearall
}
----------------------------------
----------------------------------
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
tht_xbffc.nc | cut -c15- > tht
read tht < tht
sleep 1s
rm -f xbffc.astart tht_xbffc.nc
tht
echo "Given Initial conditions
are:"
echo "Year is=" $year
echo "Month is=" $mon
echo "Starting day is=" $day
```

```
echo "Terrain height extracted is
(m) =" $tht
echo "Latitude (deg) =" $lat
echo "Longitude (deg)  =" $lon

## -------------------------------
---- PART 1.............Extracting
NCUM forecast xbjgc.pp0 and
umglaa_pe024.pp0 files
##Extracting the UMforecast *.pp0
files and converting into
regulargrid
echo "------------------ "
echo "PART 1 is running: "
echo "------------------ "
echo "Extracting the UMforecast
*.pp0 files and converting into
regulargrid (INDIA) "
echo "This process repeated for 3
continuous day on all cycles (00,
06, 12, 18) from the given date: "
echo "Each day will add to initial
day..."
for i in {0..2};
do
 #echo $i > dayadd
 #echo $day > day
 #cat day dayadd | awk '{sum+=$1}
END {print sum}' > daysum
 #read j < daysum
 #rm -f day dayadd daysum
 j=$day
 if [ $j -lt 10 ]
 then
  rday=0$j
 else
  rday=$j
 fi
  ##date=$year$mon$rday
  ##date1=${year}_${mon}_${rday}
  ./advday /$rday$mon$year +$i
yyyymmdd > fdate
  ./advday /$rday$mon$year +$i
yy_mm_dd > sdate
  read date < fdate
  read date1 < sdate
 for cyc in {00,06,12,18}
 do
 ##echo ${date} 00${cyc} >>
dates.dat
 ##cp
/gpfs3/home/umfcst/NCUM/fcst/${dat
e}/${cyc}/qwqg00.pp0
qwqg_${date1}_${cyc}.pp0  ##11
timesteps
 cp
$UMDIR/${date}/${cyc}/xbjgc.pp0
qwqg_${date1}_${cyc}.pp0     ##1
time-step
 du -k qwqg_${date1}_${cyc}.pp0 |
cut -f1 > fsize
 read ssize < fsize
 if [ $ssize -eq 0 ]
 then
   echo "The file size is zero: "
qwqg_${date1}_${cyc}.pp0
   echo "Not consider that file
for SCM forecast to avoid NRECS
problem in Genesis.scm"
 else
 echo ${date} 00${cyc} >>
dates.dat
 #$TCLDIR/subset.tcl -i
qwqg_${date1}_${cyc}.pp0 -o
var${date}_${cyc}.nc -of netcdf -
xs 0 -xe 360 -xi 0.25 -ys -90 -ye
90 -yi 0.25
 echo "Extracting only the INDIAN
domain data with 0.25
resolution...."
 $TCLDIR/subset.tcl -i
qwqg_${date1}_${cyc}.pp0 -o
var${date}_${cyc}.nc -of netcdf -
xs 60 -xe 100 -xi 0.25 -ys 0 -ye
40 -yi 0.25
 cdo seltimestep,1
var${date}_${cyc}.nc
analvar${date}_${cyc}.nc
 fi
 rm -f fsize ssize
 done
  rm -f fdate sdate

## Extracting the umglaa_pe024
files on each day at 00 UTC only
 echo "Extracting the umglaa_pe024
files:"
 du -k
$UMDIR/${date}/00/umglaa_pe024 |
cut -f1 > fsize1
 read ssize1 < fsize1
 if [ $ssize1 -eq 0 ]
 then
   echo "Umglaa_pe024 file is not
extracted since size is zero"
   echo "so it will avoid the data
in grads file"
 else
 ln -fs
$UMDIR/${date}/00/umglaa_pe024 .
convshR81.91 globalpars.tcl
umglaa_pe024
----------------------------------
----------------------------------
set outformat netcdf
set filetype 0
#  umglaa_pe024 files extractions
3-u@10m_bgrid, 4-v@10m_bgrid, 6-
temp@1.5m;
set fieldlist1 "2 3 5"
foreach infile1 [glob
umglaa_pe024] {
```

```
   set outfile1 "globalpars_[file
tail [file rootname $infile1].nc]"
   readfile $filetype $infile1
   writefile $outformat $outfile1
$fieldlist1
   clearall
}
----------------------------------
----------------------------------
 mv globalpars_umglaa_pe024.nc
globalpars_umglaa_pe024_${date}.nc
 rm -f umglaa_pe024
done

##Combining all .nc regular grid
files (4cycles per day) into one
.nc file and split one .nc file as
per variable name....
##cdo mergetime
analvar${date}_**.nc
analysisfiles_new.nc
cdo cat analvar${year}**.nc
analaysisfiles_new.nc
rm -f *.pp0 analvar**.nc var**.nc
cdo splitvar analaysisfiles_new.nc
f
##ncks -v -u analaysisfiles_new.nc
u.nc
rm -f fp.nc fdz_dt.nc fht.nc

##Combining all the umglaa_pe024
files on all selected days into
one
echo "combining and converting the
umglaa_pe024, temp, u, v files
into one and to grads:"
cdo cat
globalpars_umglaa_pe024_**.nc
globalpars_umglaa_pe024.nc
convshR81.91 globalparsoutput.tcl
umglaa_pe024.nc
----------------------------------
----------------------------------
set outformat grads
set filetype 0
set fieldlist1 "0 1 2"
#  Read in each of the
xbffc.astart, surface files and
(u,v,temp)
foreach infile1 [glob
globalpars_umglaa_pe024.nc] {
   set outfile1 "[file tail [file
rootname $infile1]]"
   readfile $filetype $infile1
   writefile $outformat $outfile1
$fieldlist1
   clearall
}
----------------------------------
----------------------------------
rm -f
globalpars_umglaa_pe024_**.nc
cdo splitvar
globalpars_umglaa_pe024.nc g
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
gu.nc > umet_${tdate}
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
gv.nc > vmet_${tdate}
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
gtemp.nc > tmet_${tdate}

## ------------------------------
---- PART 2...............GENESIS
preparation files.inp base.inp
dates.dat
echo "PART 2 is running: "
echo "Genesis preparation creaing
dates.dat already created "
echo "Genesis preparation creaing
files.inp "
rm -f base.inp gen.sh files.inp
ls f*.nc >> files
sed '2!d' files > aa
sed '1!d' files > bb
sed '5!d' files > cc
sed '6!d' files > dd
sed '4!d' files > ee
sed '3!d' files > ff
cat aa bb cc dd ee ff > files.inp
rm -f aa bb cc dd ee ff files

##Creating a base.inp file to run
the genesis from master file
echo "Genesis preparation creaing
base.inp "
cp base_master.inp a1
sed -e 's/terrainheight/'$tht'/g'
a1 > b1
sed -e "s/intdate/$(head -1
dates.dat| cut -c3-8)/g" b1 > c1
sed -e "s/findate/$(tail -1
dates.dat| cut -c3-8)/g" c1 > d1
sed -e "s/inthr/$(head -1
dates.dat| cut -c12-13)/g" d1 > e1
sed -e "s/finhr/$(tail -1
dates.dat| cut -c12-13)/g" e1 > f1
sed -e 's/intyear/'$year'/g' f1 >
g1
sed -e 's/intmonth/'$mon'/g' g1 >
h1
sed -e 's/intday/'$dayy'/g' h1 >
i1
sed "s@directory@$PWD@" i1 >
base.inp
rm -f a a1 b1 c1 d1 e1 f1 g1 h1 i1
```

```
##NOW 3 input files are created
files.inp, base.inp, dates.dat
(Check manually if you have
doubts)
######----------------------------
---PART 3 ------------------------
----running GENESIS program need
template.scm also as a one of
input
echo "PART 3 is running: "
cat dates.dat | wc -l > times
read nt < times
echo $nt
echo $lon
echo $lat
echo $loc
echo "Running the genesis_san70
creasted by asandeep (2 mins)..."
#./genesis_san70 -d -t${nt} -l18 -
x1441 -y721 -X${lon} -Y${lat} -D -
R -U < base.inp > outlog${loc}.txt
./genesis_san70 -d -t${nt} -l18 -
x161 -y161 -X${lon} -Y${lat} -D -R
-U < base.inp > outlog${loc}.txt
sleep 10
rm -f times genesis.scm
charney.scm nbtimes

#####-----------------------------
-PART 4-----------------extracting
xbffc.astart files from last day
00 cycle and
##Retrieving the required
parameters from that file and
putting into namelist.scm
##date=20160503
echo "PART 4 is running: "
echo "The Last date is to run
part4: " $date
ln -fs
$UMDIR/$date/00/xbffc.astart .
echo "Extracting the required
surface conditions from
xbffc.astart of the last day file:
"
convshR81.91 scmfields.tcl
xbffc.astart
----------------------------------
----------------------------------
set outformat netcdf
set filetype 0
#  RUNDATA@@ 17-tstar_land; 18-zh;
25-orog; 37-tstar_sice; 49-
albsoil; 69-ntml
set fieldlist1 "17 18 25 37 49 69"
#  JULES@@  52-canopy; 53-catch;
54-tstar_tile; 55-z0_tile; 60-
snow_tile
set fieldlist2 "52 53 54 55 60"
#  INPROF@@   11-sil_orog_land;
12-ho2r2_orog; 15-canopy_gbi; 16-
snodepi; 17-tstari; 19-z0m_seai;
23-ice_fract; 24-di
set fieldlist3 "11 12 15 16 17 19
23 24"
##Level extraction ALL
#  5-smcli; 13-t_deep_soili; 38-
Ozone 46-fract_typ; 47-lai; 48-
canht
set fieldlist4 "5 13 38 46 47 48"
#  Read in each of the
xbffc.astart, surface files and

foreach infile1 [glob
xbffc.astart] {
   set outfile1 "rundata_[file
tail [file rootname $infile1].nc]"
   readfile $filetype $infile1
   writefile $outformat $outfile1
$fieldlist1
   clearall
}

## JULES parameter extraction
foreach infile2 [glob
xbffc.astart] {
   set outfile2 "jules_[file tail
[file rootname $infile2].nc]"
   readfile $filetype $infile2
   writefile $outformat $outfile2
$fieldlist2
   clearall
}

## INPROF data extraction
foreach infile3 [glob
xbffc.astart] {
   set outfile3 "inprof_[file tail
[file rootname $infile3].nc]"
   readfile $filetype $infile3
   writefile $outformat $outfile3
$fieldlist3
   clearall
}

## Different levels extraction
from all filelds of scm
namelist.scm
foreach infile4 [glob
xbffc.astart] {
   set outfile4 "levs_[file tail
[file rootname $infile4].nc]"
   readfile $filetype $infile4
   writefile $outformat $outfile4
$fieldlist4
   clearall
}
----------------------------------
----------------------------------
sleep 10
./scmfields.sh
```

```
------------------------------------
------------------------------------
## Shell script for extrating
u,v,t,q,z,mslp from um analysis ##
set -x
#lat=26.43
#lon=80.33
rm -f rundata jules inprof sm st
frac_typ lai canht ozonelev
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
rundata_xbffc.nc > rundata
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
jules_xbffc.nc > jules
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
inprof_xbffc.nc > inprof

cdo selvar,sm levs_xbffc.nc sm.nc
cdo selvar,soiltemp levs_xbffc.nc
st.nc
cdo selvar,field1391 levs_xbffc.nc
frac_typ.nc
cdo selvar,field1392 levs_xbffc.nc
lai.nc
cdo selvar,field1393 levs_xbffc.nc
canht.nc

cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
sm.nc > sm
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
st.nc > st
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
frac_typ.nc > frac_typ
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
lai.nc > lai
cdo -outputtab,lon,lat,value -
remapnn,lon=${lon}_lat=${lat}
canht.nc > canht
rm -f sm.nc st.nc frac_typ.nc
lai.nc canht.nc

###Extracting onzone data
cdo selvar,O3 levs_xbffc.nc
ozone.nc
seq -89.921875 0.15625 89.921875 >
latlons
awk -v c=1 -v t=$lat 'NR==1{d=$c-
t;d=d<0?-d:d;v=$c;next}{m=$c-
t;m=m<0?-
m:m}m<d{d=m;v=$c}END{print v}'
latlons | cut -c1-7 > nearlat
read nearlat < nearlat
echo "Nearest lat value is=="
$nearlat
cdo -outputtab,lon,lat,lev,value
ozone.nc | grep $nearlat | awk 'NR
% 2 == 0' | awk '{print $4}' >
ozonelev
du ozonelev | cut -c1-2 > size
read ss < size
echo "Ozone data size==" $ss
#bias=0.001
while [ $ss -le 1 ]
do
 if [ $ss -eq 0 ]
 then
  nearlat=$(( nearlat + 0.001 ))
  echo "Nearest lat value with
bias=" $nearlat
  cdo -outputtab,lon,lat,lev,value
ozone.nc | grep $nearlat | awk 'NR
% 2 == 0' | awk '{print $4}' >
ozonelev
  sleep 10s
  du ozonelev | cut -c1-2 > size
  read ss < size
 else
  echo "Ozone data is retrieved"
 fi
done
rm -f nearlat latlons ozone.nc
size latlons nearlat_bias
------------------------------------
------------------------------------
echo "Adding the extracted fields
into namelist.scm: "
./addnamelist.sh
------------------------------------
------------------------------------
## Shell script for extrating
u,v,t,q,z,mslp from um analysis ##
set -x
cp namelist.scm a1

## RUNDATA variables replacement
in namelist.scm
sed 's/\(    ntml          =
\).*/\1999.99 ,/' a1 | sed
"s/999.99/$(sed '6!d' rundata |
cut -c15-)/g" > b1
sed 's/\(    zh            =
\).*/\1999.99 ,/' b1 | sed
"s/999.99/$(sed '2!d' rundata |
cut -c15-)/g" > c1
sed 's/\(    tstar_land    =
\).*/\1999.99 ,/' c1 | sed
"s/999.99/$(sed '1!d' rundata |
cut -c15-)/g" > d1
sed 's/\(    tstar_sice    =
\).*/\1999.99 ,/' d1 | sed
"s/999.99/$(sed '4!d' rundata |
cut -c15-)/g" > e1
sed 's/\(    albsoil       =
\).*/\1999.99 ,/' e1 | sed
```

```
"s/999.99/$(sed '5!d' rundata |
cut -c15-)/g" > f1
sed 's/\(    orog          =
\).*/\1999.99 ,/' f1 | sed
"s/999.99/$(sed '3!d' rundata |
cut -c15-)/g" > g1

## JULES parameters replacement in
namelist.scm
sed 's/\(    canopy     =
\).*/\1999.99 ,/' g1 | sed
"s/999.99/$(sed '1!d' jules | cut
-c15-)/g" > h1
sed 's/\(    catch      =
\).*/\1999.99 ,/' h1 | sed
"s/999.99/$(sed '2!d' jules | cut
-c15-)/g" > i1
sed 's/\(    snow_tile  =
\).*/\1999.99 ,/' i1 | sed
"s/999.99/$(sed '5!d' jules | cut
-c15-)/g" > j1
sed 's/\(    z0_tile    =
\).*/\1999.99 ,/' j1 | sed
"s/999.99/$(sed '4!d' jules | cut
-c15-)/g" > k1
sed 's/\(    tstar_tile =
\).*/\1999.99 ,/' k1 | sed
"s/999.99/$(sed '3!d' jules | cut
-c15-)/g" > l1

## INPROF parameters replacement
in namelist.scm
sed 's/\(    z0mseai       =
\).*/\1999.99 ,/' l1 | sed
"s/999.99/$(sed '6!d' inprof | cut
-c15-)/g" > m1
sed 's/\(    canopy_gbi    =
\).*/\1999.99 ,/' m1 | sed
"s/999.99/$(sed '3!d' inprof | cut
-c15-)/g" > n1
sed 's/\(    sil_orog_land  =
\).*/\1999.99 ,/' n1 | sed
"s/999.99/$(sed '1!d' inprof | cut
-c15-)/g" > o1
sed 's/\(    tstari        =
\).*/\1999.99 ,/' o1 | sed
"s/999.99/$(sed '5!d' inprof | cut
-c15-)/g" > p1
sed 's/\(    ho2r2_orog    =
\).*/\1999.99 ,/' p1 | sed
"s/999.99/$(sed '2!d' inprof | cut
-c15-)/g" > q1
sed 's/\(    ice_fract     =
\).*/\1999.99 ,/' q1 | sed
"s/999.99/$(sed '7!d' inprof | cut
-c15-)/g" > r1
sed 's/\(    di            =
\).*/\1999.99 ,/' r1 | sed
"s/999.99/$(sed '8!d' inprof | cut
-c15-)/g" > s1
sed 's/\(    snodepi        =
\).*/\1999.99 ,/' s1 | sed
"s/999.99/$(sed '4!d' inprof | cut
-c15-)/g" > t1

## EXTRA
sed 's/\(    smci           =
\).*/\1999.99 ,/' t1 | sed
"s/999.99/$(sed '1!d' sm | cut -
c15-)/g" > u1
## LEVEL data processing and
adding into namelist.scm----JULES
sed 's/\(    smcli       =
\).*/\1999.99 ,/' u1 | sed
"s/999.99/$(awk '{ print $3 }' sm
| xargs | sed -e 's/ /, /g')/g" >
v1
###sed 's/\(    frac_typ   =
\).*/\1999.99 ,/' v1 | sed
"s/999.99/$(awk '{ print $3 }'
frac_typ | xargs | sed -e 's/ /,
/g')/g" > w1
cp v1 w1
sed 's/\(    canht       =
\).*/\1999.99 ,/' w1 | sed
"s/999.99/$(awk '{ print $3 }'
canht | tail -3 | xargs | sed -e
's/ /, /g')/g" > x1
sed 's/\(    lai         =
\).*/\1999.99 ,/' x1 | sed
"s/999.99/$(awk '{ print $3 }' lai
| xargs | sed -e 's/ /, /g')/g" >
y1
## LEVEL data processing and
adding into namelist.scm----INPROF
sed 's/\(    t_deep_soili   =
\).*/\1999.99 ,/' y1 | sed
"s/999.99/$(awk '{ print $3 }' st
| xargs | sed -e 's/ /, /g')/g" >
z1

## OZONE processing
sed 's/\(    ozone          =
\).*/\1999.99 ,/' z1 | sed
"s/999.99/$(cat ozonelev | xargs |
sed -e 's/ /, /g')/g" >
namelistcorr.scm
rm -f a1 b1 c1 d1 e1 f1 g1 h1 i1
j1 k1 l1 m1 n1 o1 p1 q1 r1 s1 t1
u1 v1 w1 x1 y1 z1

## Run1-317,296,do3:: Run2-
275,254,do3:: Run3-254,233:: Run4-
follows
##17 August 2016 changed 221 to
223.... 242 to 244...and also
added sed '205d' b2 >
c2....(because of / added at
INPROF section...i.e. due to
moified template.scm."""")
```

```
## 17 aug 2016...13.10time
...again re-modiefied to older
version....
cp namelistcorr.scm
namelistcorrected.scm
#sed '244d' namelistcorr.scm > a2
#sed '223d' a2 > b2
#sed '205d' b2 > c2
##18 aug 2016...modified
template.scm to / in INPROF
section...so modiefieed firtst
three below lines.
sed '243d' namelistcorr.scm > a2
sed '222d' a2 > b2
sed '204d' b2 > c2
sed '98d' c2 > d2
sed '94d' d2 > e2
sed '59d' e2 > f2
sed '21d' f2 > namelistfinal.scm
rm -f a2 b2 c2 d2 e2 f2
rm -f rundata jules inprof sm st
frac_typ lai canht ozonelev
----------------------------------
----------------------------------
rm -f latlons size nearlat

#####-----------------------------
PART 5--------------------Running
the SCM on BHASKARA ONCE THE scm
job compilation over
##Check carefully whether scm job
is compied properly or not SIZES,
CNTLATM, scmjob.lsf, and copy the
namelist.scm into that
##path and run bsub < scmjob.lsf
##Sending the namelist file to the
JOB dir and running the forecast
for 3 days.
echo "PART 5 is running: "
echo "Running the SCM on BHASKARA
ONCE THE scm job compilation over:
"
rm -f $SCMDIR/namelist.scm out.log
err.log NCUM_SCMoutput.nc
cp namelistfinal.scm
namelistfinal_$tdate.scm
mv namelistfinal.scm
$SCMDIR/namelist.scm
cd $SCMDIR
echo "SCM Job is submitting: "
bsub < scmjob.lsf
----------------------------------
----------------------------------
##!/usr/bin
#file to run $RUNID.exec
#BSUB -a poe
#BSUB -J scmjob.lsf
#BSUB -o
/gpfs2/home/asandeep/umui_runs/xbq
la-071102419/out.log
#BSUB -e
/gpfs2/home/asandeep/umui_runs/xbq
la-071102419/err.log
#BSUB -n 1
#BSUB -W 01:00
#BSUB -q small
#BSUB -x
#JOBDIR=/gpfs2/home/asandeep/umui_
runs/xbqla-071102419/
export VN=8.6

export
UNIT57=/gpfs1/home/moum/UM/vn8.6/c
tldata/spectral/ga3_1/sp_sw_ga3_0
export
UNIT80=/gpfs1/home/moum/UM/vn8.6/c
tldata/spectral/ga3_1/sp_sw_ga3_0
export
JOBDIR=/gpfs2/home/asandeep/umui_r
uns/xbqla-071102419

procs 1 -hfile hostfile
mpirun.lsf
/gpfs2/home/asandeep/umui_runs/xbq
la-071102419/xbqla.exe
----------------------------------
----------------------------------
sleep 50
cp NCUM_SCMoutput.nc $RUNDIR
rm -f NCUM_SCMoutput.nc
namelsit.scm out.log err.log
domain.dat scumlist.dat
cd $RUNDIR

#####-------------------------
PART 6----------------------
Extracting the required fields
from SCM and Global model forecast
echo "PART 6 is running: "
echo "Extracting the required
fields from SCM output and
converting into GrADS format"
convshR81.91 scmoutput.tcl
NCUM_SCMoutput.nc
----------------------------------
----------------------------------
set outformat grads
set filetype 0
#  METPARS@@ 237-RH@1.5m; 240-
WS@10m; 241-WD@10m; 249-T@1.5m
set fieldlist1 "237 240 241 249"
#  RADIATION BUDGET@@  22-
SWR@surface; 48-LWF@surface; 243-
LHF@surface, 302-SHF@surface
set fieldlist2 "22 48 243 302"
foreach infile1 [glob
NCUM_SCMoutput.nc] {
   set outfile1 "metpars_[file
tail [file rootname $infile1]]"
   readfile $filetype $infile1
```

```
   writefile $outformat $outfile1
$fieldlist1
   clearall
}

## Radiation parameter extraction
from NCUM_SCMoutput.nc
foreach infile2 [glob
NCUM_SCMoutput.nc] {
   set outfile2 "radpars_[file
tail [file rootname $infile2]]"
   readfile $filetype $infile2
   writefile $outformat $outfile2
$fieldlist2
   clearall
}
----------------------------------
----------------------------------
echo "the ctl file modified date
is:" $cdate
echo "correcting the date in the
ctl files for metpars and
radpars:"
echo "converting corrected ctl
file into netcdf data:"
sed '11 a\tdef 288 linear 00:cdate
15mn' metpars_NCUM_SCMoutput.ctl >
grad1.ctl
sed -e '11d' grad1.ctl >
metpars_NCUM_SCMoutput_corr.ctl
sed -i '11s/cdate/'$cdate'/g'
metpars_NCUM_SCMoutput_corr.ctl
cdo -f nc import_binary
metpars_NCUM_SCMoutput_corr.ctl
metpars_NCUM_SCMoutput_corr_${tdat
e}.nc

sed '11 a\tdef 288 linear 00:cdate
15mn' radpars_NCUM_SCMoutput.ctl >
grad2.ctl
sed -e '11d' grad2.ctl >
radpars_NCUM_SCMoutput_corr.ctl
sed -i '11s/cdate/'$cdate'/g'
radpars_NCUM_SCMoutput_corr.ctl
cdo -f nc import_binary
radpars_NCUM_SCMoutput_corr.ctl
radpars_NCUM_SCMoutput_corr_${tdat
e}.nc
rm -f grad1.ctl grad2.ctl


####--------------------------
PART 7----------------------output
and plotting through GrADS
echo "Plotting using GrADS script
metpars and radpars forecasting
for 3 days"
sed -e 's/gdate/'$tdate'/g'
metpars_master.gs >
metpars${loc}old.gs
sed -e 's/location/'$loc'/g'
metpars${loc}old.gs >
metpars$loc.gs
sed -e 's/gdate/'$tdate'/g'
radpars_master.gs >
radpars${loc}old.gs
sed -e 's/location/'$loc'/g'
radpars${loc}old.gs >
radpars$loc.gs
sed -e 's/gdate/'$tdate'/g'
meteogram_global_master.gs >
met1.gs
sed -e 's/location/'$loc'/g'
met1.gs > met2.gs
sed -e 's/latt/'$lat'/g' met2.gs >
met3.gs
sed -e 's/long/'$lon'/g' met3.gs >
meteogram_global$loc.gs

grads -pc metpars$loc.gs
grads -pc radpars$loc.gs
grads -pc meteogram_global$loc.gs

rm -f outlog$loc.txt
metpars$loc.gs radpars$loc.gs
metpars${loc}old.gs
radpars${loc}old.gs met1.gs
met2.gs met3.gs
meteogram_global$loc.gs
mv metpars_Kanpur.png
metpars_Kanpur_${tdate}.png
mv radpars_Kanpur.png
radpars_Kanpur_${tdate}.png
mv metpars_global_Kanpur.png
metpars_global_Kanpur_${tdate}.png
echo "Two png files are created 1)
Meteorological parameters
(T,RH,WS,WD) 2) Radiation
parameters (SWR,LWR,SHF,LHF)"
echo "Thrid png file is from
Global NCUM forecasting
umglaa_pe024 files data temp,ws"
echo "An corrected ctl file
generated nc was created for the
post processing with othe gui's:"
echo "End for the current date"

####-------------------PART 8-----
-----------copying the outputfiles
into mkdir ${loc}_${mont}${year}
echo "------------------ "
echo "PART 8 is running: "
echo "------------------ "
mv umet_${tdate} vmet_${tdate}
tmet_${tdate}
metpars_${loc}_${tdate}.png
radpars_${loc}_${tdate}.png $OPDIR
mv
metpars_NCUM_SCMoutput_corr_${tdat
e}.nc
```

```
radpars_NCUM_SCMoutput_corr_${tdat
e}.nc $OPDIR
mv
metpars_global_${loc}_${tdate}.png
namelistfinal_${tdate}.scm $OPDIR
mv analaysisfiles_new.nc
$OPDIR/analaysisfiles_${tdate}.nc
## File size checking (below)
access only if it.
fi
rm -f filesize
##End of the month (below)
done
echo "SCM Runs completed
successfully for all days:"
echo "Clear the RUNDIR which are
not useful: After all days are
completed"
rm -f f*.nc files.inp base.inp
dates.dat genesis.scm charney.scm
namelist.scm analaysisfiles_new.nc
rm -f rundata_xbffc.nc
jules_xbffc.nc inprof_xbffc.nc
levs_xbffc.nc xbffc.astart
namelistcorr.scm namelistfinal.scm
ozone.nc
rm -f NCUM_SCMoutput.nc
metpars_NCUM_SCMoutput.ctl
metpars_NCUM_SCMoutput.dat
radpars_NCUM_SCMoutput.dat
radpars_NCUM_SCMoutput.ctl
rm -f tht tht_xbffc.nc
metpars_NCUM_SCMoutput_corr.ctl
radpars_NCUM_SCMoutput_corr.ctl
metpars_NCUM_SCMoutput_corr.nc
radars_NCUM_SCMoutput_corr.nc
rm -f gu.nc gv.nc gtemp.nc
globalpars_umglaa_pe024.nc
globalpars_umglaa_pe024.ctl
globalpars_umglaa_pe024.dat

END=$(date +%s)
DIFF=$(( $END - $START ))
echo "It took $DIFF seconds"
echo "*********----------THE END--
-------**********"
```

Appendix B: List of parameters in namelist.scm tuned to Kanpur

Variable	Description	Default/ Change
nfor	No. of observation forcing time-levels for each forcing variable in namelist &inobsfor in file namelist.scm	12
model_levels_nml	No. of model levels in variable profiles in namelist.scm. This must match model_levels in control file SIZES	70
l_ts_log	Turns on timestep message, gives use an indication of run progress	True
land_points	Number of land points in run. This must be consistent with land_sea_mask	1
year_init	Starting year (yyyy)	2016
month_init	Starting month (mm)	04
day_init	Starting day (dd)	19
hour_init	Starting hour (hh)	00
min_init	Starting minute (mm)	00
sec_init	Starting second (ss)	00
Lat	Latitude of SCM run (dd.rr)	26.44
Lon	Longitude of SCM run (dd.rr)	80.33
soil_type	1 (Ice), 2 (Fine), 3 (Medium), 4 (Coarse)	2
gridbox_area	Global dimensions (m^2)	2.6e+6
exname_in	Experiment name (tape input)	NSCM
exname_out	Experiment name (tape output)	NSCM
ndayin	Days requested to run	3
nminin	Minutes requested to run	0
nsecin	Seconds requested to run	0
timestep (obs_pd)	Model timestep for physics routines (seconds)	900

ntrad1	First timestep on which radiation called	1
ntml	Surface mixed layer, top level	1
zh	Height (above srf.) at top of boundary layer (m)	75.57
tstar_sea	Open sea surface temperature (K)	
tstar_land	Land mean surface temperature (K)	295.54
tstar_sice	Sea-ice surface temperature (K)	
fland_ctile	Land fraction on land points	1
albsoil	Soil albedo (0-1)	0.24
orog	Orography height (m)	124.8
ozone	Ozone profile (kg/kg)	θ levels
ancyc	Use annual cycle for varying radiation input	True
altdat	Use specified initial profiles of ϴ, q, u and v from &INPROF	False
land_sea_mask	Mask for land points	True
land_ice_mask	Mask for land-ice points	False
soil_mask	Mask for land-soil points	True
obs	Use large-scale observational forcing	True
l_spec_z0	Use prescribed roughness lengths	False
geoforce	Apply geostrophic wind forcing	True
geoinit	Initialise dump to geostrophic.	False
grafdump_step	Graphic dump of mean values required each dump_step	False
local_time	User local time rather than GMT for diagnostics	False
stats	Use large-scale statistical forcing	False
obs_surf	Use surface forcing	True
prindump_step	Printout of mean daily dump required each dump_step	False
prindump_day	Printout of mean daily dump required	False
gs	Stomatal conductance 1.0	0.01
canopy	Surface/canopy water (snow-free land tiles) (Kg/m^2)	9*0.0
rgrain	Snow grain size (μm)	9*50.0
smi_opt	Option to set method of initialising soil mositure content 0:Use smcli, 1:Use fsmc, 2:Use sth	0
smcli	Initial profile of soil moisture content in soil layers (kg/m^2)	1.89791, 59.5739, 178.111, 631.442,
frac_typ	Fractions of surface types	2*0.0, 1.00. 6*0.0
canht	Canopy height (m)	1.4605, 1.25992, 1.5874 ,
catch	Surface/canopy water capacity (snow-free land tiles) (kg/m^2)	0.42
snow_tile	Lying snow on tiles (0.0 m)	9*0.0
lai	Leaf area index	0.306255, 0.271866, 0.309315,
z0_tile	Tile roughness lengths (m)	0.16
infil_tile	Maximum surface infiltration	4.00, 4.00, 2.00, 2.00,

		2.00, 5*0.0
tstar_tile	Surface tile temperature. Initialised to tstari from &INPROF if not set (K)	295.54
theta	Initial potential temperature profile (θ-levels) (K)	θ-levels
qi	Initial specific humidity profile (θ-levels) (kg/kg)	θ-levels
p_in	Pressure profile on ρ-levels(ρ-levels) (Pa)	ρ-levels
ui	Initial zonal wind profile (ρ-levels) (m/s)	ρ-levels
vi	Initial meridional wind profile (ρ-levels) (m/s)	ρ-levels
wi	Initial vertical wind profile (θ-levels) (m/s)	θ-levels
u_0	Zonal component of surface current (m/s)	0.0
v_0	Meridional component of surface current (m/s)	0.0
tstari	Initial surface temperature (K)	296.1
z0m_seai	Initial sea surface roughness length (m)	6.6e-4
smci	Initial soil moisture content (kg/m^2)	1.87
canopy_gbi	Initial gridbox mean canopy water (kg/m^2)	0.0
t_deep_soil	Initial deep soil temperatures (K) (5, 10, 20, 50) cm	300.615, 302.948, 301.203, 297.272 ,
snodepi	Initial snow depth (kg/m^2)	0.0
ice_fract	Gridbox sea-ice fraction	0.0
sil_orog_land	Silhouette area of unresolved orography per unit horizontal area (land points)	0.001
ho2r2_orog	horizontal area (land points) Real (rw,rl) ho2r2_orog Standard deviation of orography, equivalent to peakto-trough height of unresolved orography divided by 2 √ 2 (land points) (m)	2.18
nml_inprof_thetal	Option for interpretation of theta and qi profiles when initialising qcl/qcf	0
di	Equivalent sea-ice thickness (m)	0.0
z_tom_nml	Top of the model (m)	40000.0
eta_th_nml	Theta profile	θ-levels
eta_rh_nml	Rho profile	ρ-levels
ug_opt	Geostrophic zonal wind at surface (m/s)	0
vg_opt	Geostrophic meridional wind at surface (m/s)	0
ug	Geostrophic zonal wind (m/s)	5
vg	Geostrophic meridional wind (m/s)	5
t_inc	Temperature increments (K/day) 1:nfor*nlev	θ-levels
q_star	Specific humidity increment (kg/kg/day) 1:nfor*nlev	θ-levels
u_inc	Zonal wind increment (m/s/day) 1:nfor*nlev	ρ-levels
v_inc	Meridional wind increment (m/s/day) 1:nfor*nlev	ρ-levels
w_inc	Verical wind increment (m/s/day) 1:nfor*nlev	ρ-levels
old_vertadv	Tempeartue increment for vertical advection	False
l_vertadv	Enable the vertical advection	False
rlx_t	Use forcing relaxation: Temperature	1
rlx_q	Use forcing relaxation: Specific humidity	1
rlx_u	Use forcing relaxation: Zonal wind	1
rlx_v	Use forcing relaxation: Meridional wind	1
rlx_w	Use forcing relaxation: Vertical velocity	1

References:

Davies, L., Jakob, C., Cheung, K., Del Genio, A., Hill, A., Hume, T., Keane, R. J., Komori, T., Larson, V.E., Lin, Y., Liu, X., Nielsen, B. J., Petch, J., Plant, R. S., Singh, M. S., Shi, X., Song, X., Wang, W., Whitall, M.A., Wolf, A., Xie, S., and Zhang, G.: A single-column model ensemble approach applied to the TWP-ICE experiment, *J. Geophys. Res. Atmos.*, 118, no. 12, 6544-6563, doi:10.1002/jgrd.50450, 2013.

Rajagopal, E. N., Iyengar, G. R., George, J. P., Gupta, M. D., Mohandas, S., Siddharth, R., Gupta, A., Chourasia, M., Prasad, V. S., Aditi, Sharma, K., and Amit, A.: Implementation of Unified Model based analysis-forecast system at NCMRWF, NMRF/TR/2/2012, PP 1-45, 2012.

Randall, A.: Single column models as a bridge between observations and climate models, ARM Science meeting, 1994.

Randall, D.A., and Cripe, D.G.: Alternative methods for specification of observed forcing in single-column models and cloud system models, J. Geophys. Res., 104 (D20):24, 527–545, 1999.

Wong, R., The single column model, Documentation Paper C9, Met Office, PP 1-57, 2010.