The Neelin-Zeng Quasi-Equilibrium Tropical Circulation Model (QTCM1) Version 2.3

J. David Neelin
Ning Zeng
Chia Chou
Johnny Lin
Hui Su
Matthias Munnich
Katrina Hales
Joyce Meyerson

Department of Atmospheric Sciences
University of California, Los Angeles

September 26, 2002

Contents

Preface

This manual describes the single baroclinic mode version of the Neelin-Zeng Quasi-Equilibrium Tropical Circulation Model (QTCM1), developed at the Department of Atmospheric Sciences at the University of California, Los Angeles (UCLA). Primary development of the QTCM1 was done by J. David Neelin and Ning Zeng. The original driver was written by William Weibel. Alistair Adcroft provided the direct Poisson solver used in the QTCM1. Chia Chou developed a cloud/radiation package used in the model. Hui Su and Ning Zeng wrote the slab mixed-layer ocean module. Bjorn Stevens led the development of the atmospheric boundary layer (ABL). Matthias Munnich implemented the ABL and converted the code to Fortran 90. This manual was written by Johnny Lin, Ning Zeng, David Neelin, Chia Chou, Hui Su and Matthias Munnich. Contributions to the documentation analysis and testing were also made by Katrina Hales and Joyce Meyerson.

This model and its documentation is, by necessity, offered to the community on an "as-is" basis, with no warranties or guarantees given or implied. However, we earnestly desire comments and suggestions to help improve both the model and the manual. Users are particularly requested to please notify us of any bugs they find in the model. The QTCM1 development team can be reached in care of:

Prof. J. David Neelin
University of California, Los Angeles
Department of Atmospheric Sciences
7127 Math Sciences
Box 951565
Los Angeles, CA 90095-1565
USA
Tel: (310) 825-1217
Fax: (310) 206-5219

The QTCM1 home page on the World Wide Web is located at:

`http://www.atmos.ucla.edu/~csi/QTCM/`

Links to download the model and view the online manual are also located on that page. Addenda to this manual are found on the "manual" page of the QTCM website.

The authors of the manual can be reached via e-mail at:

Hui Su	hui@atmos.ucla.edu
Ning Zeng	zeng@atmos.umd.edu
Matthias Munnich	munnich@atmos.ucla.edu
J. David Neelin	neelin@atmos.ucla.edu
Johnny Lin	air_jlin@yahoo.com

This model is distributed to the research community for non-profit research, subject to the usual academic conventions for copyright and citation. If you have benefited from this work, please cite the major papers describing this model:

Neelin, J. D., and N. Zeng, 2000: A quasi-equilibrium tropical circulation model—formulation, *J. Atmos. Sci.*, 1741-1766.

Zeng, N., J. D. Neelin, and C. Chou, 2000: A quasi-equilibrium tropical circulation model— implementation and simulation, *J. Atmos. Sci.*, 1767-1796.

along with any of the applications papers (listed at `http://www.atmos.ucla.edu/~csi`) that you have benefited from.

This work was supported in part by National Science Foundation grant ATM-0082529 (originally by ATM-9521389) and National Oceanographic and Atmospheric Administration grants number NA166P2003 (originally by NA86GP0314). Please see papers and web pages for acknowledgement of other support by outside contributors and group members who have moved on to other affiliatons.

Chapter 1

Basic Operation

1.1 Introduction and Model Description

1.1.1 Overview

The Neelin-Zeng Quasi-Equilibrium Tropical Circulation Model (QTCM1), is an intermediate complexity atmospheric model, designed to occupy the niche between general circulations models (GCMs) and simple models. The model formulation makes use of the constraints placed on the flow by deep convection, as represented by quasi-equilibrium (QE) convective parameterizations, hence the Q in QTCM. Analytical solutions for tropical flow [8, 7] under strict QE conditions (plus additional approximations) appeared useful enough to motivate development of a model that could employ these where appropriate and yet not be completely bound by them, as discussed below. The derivation from the primitive equations and the equations for the resulting atmospheric model are given in [1]. The numerical implementation developed at the UCLA Department of Atmospheric Sciences, initially by Neelin and Zeng and now by other members of the Climate Systems Interactions group, including H. Su, C. Chou and M. Munnich is more fully discussed in [1, 2]. In order to consistently simulate the tropical climatology, interannual variability and intraseasonal variability, a suite of physical parameterizations has been included, including a simple land model and a radiation package, all aimed at being simpler to run and analyze than a GCM, and yet being a reasonable approximation under suitable conditions to what would be included in a GCM. This manual describes the implementation of the resulting numerical model.

Regarding the atmospheric dynamics, QE convective closures and model formulation: QE convective closures posit that moist convective elements tend to act quickly compared to the large scale flow, and tend to dissipate the buoyancy available to moist overturning motions within a vertical column. Convective available potential energy (CAPE) is one measure of this. The tendency to generate CAPE at large-scales is assumed to be roughly balanced by dissipation in small-scale convective activity to establish a statistical "quasi-equilibrium" among variables affecting buoyancy. Typically, this implies a relation between the temperature profile through the column and the boundary layer moisture and temperature. One parameterization that makes this explicit in a simple manner is the Betts-Miller convective scheme. A convective profile is established by a moist adiabat arising from the boundary layer. Temperature is restored toward this profile when deep convection occurs. In moist convective regions, temperature will thus tend to have a vertical profile close to this convective profile.

QTCM1 takes the shape of this convective temperature profile and uses it as a basis function for temperature in a Galerkin-like representation of vertical structure. A basis function for baroclinic velocity is computed from the implied vertical profile of the baroclinic pressure gradients, according to the hydrostatic equation. And the implied vertical velocity profile is computed from this according to continuity. Barotropic motions add an additional basis function. The primitive equations are projected on these vertical structures. In deep convective regions, this should be an efficient basis set to capture much of the most important motions, compared to a GCM running the same convection scheme. Away from deep convective regions, QE will not apply, but the Galerkin-like representation of vertical structure still tends to capture deep motions for the dry dynamics. Within a Rossby radius of deformation (25 degrees or so) this should still be a reasonable approximation. At higher latitudes, it is simply a highly truncated Galerkin representation.

The reduction of the vertical degrees of freedom considerably cuts computational time. The current version takes approximately 5 minutes of CPU time to run one year of model simulation on a Sun Ultra 80 workstation (1.5 minutes on a Pentium-4/Linux workstation). A GCM with only a few layers would have the same speed, but an advantage of the current approach is that it is more quantitative in and near deep convection zones. For tropical climate, this appears to be quite significant. In analyzing the results, it can be helpful to consider the moist static energy budget and the gross moist stability, as discussed in [1, 11].

When using the model, please note that there are trade-offs to the approach. It is always necessary to consider for each application whether the approximations are within their range of validity. Often this will have to be tested for the particular phenomenon of interest. We also note that the model does not assume strict QE, although it is set up to be most accurate when QE tends to apply. Although we find QE assumptions can be useful for some scales and phenomena, this does not mean that it will apply in all situations. In particular, QE assumes that an ensemble of convective elements is acting together. At sufficiently small scales this assumption will cease to hold well. With these caveats, we note that there do seem to be a variety of phenomena for which the model can be useful to the climate research community.

The current version of the model includes the following major features:

- Two vertical components (one barotropic and one baroclinic) of the general circulation are represented. These components are calculated from 1000 mb to 150 mb. The "1" in "QTCM1" refers to there being only one baroclinic mode simulated by the model.

- Convective heating is parameterized by a version of the Betts-Miller [17, 18, 19] parameterization scheme.

- Land surface processes are represented by the simple scheme "SLand1" of Zeng and Neelin [14] and Zeng et al. [2].

- Two different radiation cloud packages are available in the model: (i) a simpler parameterization, clrad0 (Zeng et al. [2]) which has not been tested in v2.3 and is retained for backward compatibility; and (ii) a more complicated parameterization, clrad1, [3, 4, 5] slightly revised for v2.3. An option is provided to read certain cloud types from data.

- A simple atmospheric boundary layer model for surface winds is used to calculate all surface fluxes.

1.1.2 Version History

Additional details are found in later chapters.

2.3 Released August 2002.

  - A simple formulation for atmospheric boundary layer winds is used to compute the surface fluxes.
  - Vertical momentum advection terms are included in the momentum equations.
  - Fourth-order horizontal diffusion is used in the momentum equations in lieu of the second-order diffusion. (See Section 3.6)

- New cloud parameterization is implemented, considering random overlapping of different cloud types. The resulting total cloud fraction is ensured to be less than 1. (See Section 3.3.1).
- The top of the model is raised to 150 HPa and the vertical velocity structure of the 1^{st} baroclinic mode (V_1) is held constant above the 280 HPa level for numerical reasons (See Section 3.1).
- The model domain is extended to 78.75°S – 78.75°N.
- Corrections to spherical geometry terms are included.
- A moisture dependence is added to the gross dry stability M_s to simulate the cloud-top effect on M_s. A cap is used so that the gross moist stability would not decrease to a lower value than the minimum value typical of the reference profile region.
- The interception loss *Evapi* is capped at 50% of precipitation amount (as a numerical safety check)
- The surface pressure gradients can be diagnosed.
- Baroclinic velocity potential can be diagnosed.
- The artificial ocean boundary near the northern and southern boundary used in version 2.2 is removed and the sponge layers are not needed by default. A Fourier filter is used to damp small scale features at high latitudes.
- Available Reynolds SST has been updated from December 1998 to October 2001.
- All boundary data except the cloudiness are updated for the new grid dimensions. There are numerous gaps in the ISCCP cloudiness data at high latitude making an interpolation to the new domain difficult. Currently the cloudiness data are identical to the ones used by QTCM version 2.2. The missing high latitude are set to the zonal means of the closest existing latitude band at 58.75 N and 58.75 S, respectively.
- An option is added to use prescribed observed SST in masked regions and mixed layer ocean in non-masked regions (See Section 3.4).
- Scripts to run ensemble simulations are added (See Section 1.3.2).

Changes in code structure:

- The code is adapted to Fortran-90.

- All common blocks are replaced by modules.
- Example makefiles for different platforms are provided.
- With the simple atmospheric mixed layer the surface stress no longer relates to the resolved winds by a simple linear formula. Therefore the damping parameters $\epsilon_0, \epsilon_1, \epsilon_{01}, \epsilon_{10}$ (equations (4.33) (4.34) and (5.15) in [2]) have been replaced by the corresponding surface stress terms (equations (4.12) and (4.13) in [2]).
- Stress taux,tauy components are computed on u, v grids, respectively. Previously $\epsilon_0 u_0$ etc terms related to stress were used in the time marching on u, v grids while the arrays taux, tauy were diagnostic calculated for output on the T grid.
- The order of the indices of the cloud fraction variable `cld` was changed from "lon, lat, cld-type" to "cld-type, lon, lat".
- The output routines are rewritten. GrADS control files for the binary output files are automatically generated.
- Adding (removing) an output variable now only requires adding (removing) one subroutine call.
- The scripts to run the model are modified so that multiple runs can be made simultaneously. To this end each run is executed in a separate subdirectory of directory `proc`.
- The interval to write restart files can now be specified.
- Subroutines to diagnose surface pressure and its derivatives are added.
- When the model blows up, the instantaneous values of standard variables are output for diagnosis and the monthly means for the last month before blowing up are also written.
- The prognostic variable arrays are dimensioned as (nx,0:ny+1) to avoid out-of-bounds access of memory which led to errors on some computers (e.g., IBM-AIX).
- Added `ntoutr` and `nooutr` to the input parameters to specify time in days between restart files and days to skip before starting to write a restart file. Default write interval is one year.
- The restart file output names now include are changed to `qtcm_YYYYMMDD.restart`.

- The front-end solvers for the FATD package are in qtcm.f90 (used to be in fatdpkg.f). The old front-end solvers for Cartesian Coordinates are deleted. Users can refer to v2.2 or older versions for these or set $cos\varphi = 1$.
- New preprocessor macros are

  DELTA_CO2 Enable radiation code to use a non-s standard CO_2 level.

  TOPO Include topography effects due to barotropic divergence.

  MXL_OCEAN Use mixed layer ocean.

  BLEND_SST Use readsst in mask regions and mixed layer ocean outside.

  NO_WADV Switch off vertical advection terms.

  NO_ABL Switch off atmospheric boundary layer.

  OBSCLD Use ISCCP observed cloud cover climatology.

  SPONGES Enable sponge regions.

  CHIWIND Diagnose velocity potential using irrotational winds.

  ENSEMB_INI Create initial condition files for ensemble runs.

2.2 Released November 2000.

Changes in radiation package clrad1.F:

- For solar radiation, we use maritime aerosol type with column optical depth 0.2. Compared to version 2.1, the downward solar radiation at the surface (`FSWds`) decreases and the upward solar reflection at TOA (`FSWut`) increases because of the reflection of aerosol.
- For longwave radiation, we implement a new set of coefficients to estimate effects of carbon dioxide. It is a linear relation and it is good for double CO_2 experiments. For a case with triple and more CO_2, its accuracy will be reduced.
- We also add an effect from the forth cloud type *AsAc+CuSc*. A climatological value of cloudiness for the fourth cloud type is used and the cloud cover is absorbed into the coefficients for both longwave and solar radiation schemes.

Changes in code structure:

- The code now uses a fortran 90 compiler and its preprocessor. Preprocessor macros are used as model switches

  Possible macros are:

CPLMEAN ..... Compute mean flux values for coupling with an ocean.
LINEAR_T1C .... Use the linear calculation for reference moisture profile instead of the standard nonlinear calculation.
YEAR360 ..... A year has 360 days instead of the standard 365 days (no leap years).
FCF77 .... Do not use F90 constructs such as array operations.
REAL=real*8 Use this for global real*8 (double prec.) calculations.

- The default computation is now single precision. All internal double precision function calls are replace by their generic names. Define REAL=real*8 in the makefile to switch back to double precision.
- The default computation now uses a 365 day calendar. Set ntout (ntouti) to -30 to get monthly mean output.
- The nonlinear convective closure code has been merged into the standard qtcm.F code and is the default choice.
- To ease the output of the moist stable energy `GMq` is now an array defined in qtcm.h. and is one of the standard output variables for NetCDF output.
- If day0 is not specified in the input file driver.in the date saved in the restart file is now used to set the initial time. To this end the initial call of "Timemanager" in the driver is moved into "qtcminit".
- Major restructuring of the output routines.
- Fixed the Timemanager routine for the perpetual model. For SSTmode = "perpetual", Timemanager now sets "dayofyear" to day0 of the initial month (month0).
- Changed control of when to output mean values. i.e., `loutpmean = mod(dayofmodel-noc`
- All surface fluxes are now interpolated onto the model's temperature grid.
- The makefiles for different machines are merged. Useful compiler flags are provided in comment lines.
- The link of rdsst.F and mxlayer.F to ocean.F is removed and replaced by a new target qtcm_mxl for a mixed layer version.
- Added a NetCDF output routine. To use it one has to replace the standard makefile by makefile.netcdf. This output routine requires an f90 compiler and the netcdf library available from http://www.unidata.ucar.edu.
- The subroutines in driver.F have been split off into driver_util.F.

- The calendar routines in driver_util.F are rewritten.
- The coefficients for vertical advection of momentum (`Vwijk`) are added in qtcmpar.in; however, they are not used in the momentum calculations in this version because the corresponding climatology needs improvements.
- An artificial ocean boundary is imposed at the two outmost grid points near the north and south boundaries to reduce chances of model "blow-up" associated with numerical instability at the boundaries. Therefore, no sponges are needed near the boundaries.

2.1 Released May 1999. This is the version used in [2].

Major changes and additions from version 2.0:

- Vertical profile of b1 (the first basis function of moisture perturbation from reference state) is changed to avoid negative moisture in the upper levels (see [2] p17). Corresponding radiative coefficients in clrad1.f are updated.
- The meridional boundary of the model is stretchable up to the poles. All grids are shifted a half-grid point in the meridional direction. See Section 1.1.3 for details.
- A slab mixed-layer ocean model is implemented. See Section 3.4 for details.
- Two additional convective closures are implemented in qtcm.f.mconvct2. See Section 3.5 for details.

Minor changes from version 2.0:

- utilities.f is added. See Section 1.2 for details.
- bndry.f is modified to read multiple monthly boundary data and also to conduct time interpolation into daily data.
- FirstTime.sh is renamed spinup.sh.
- Several makefiles for different machines (SUN,CRAY and SGI) are provided. A set of shell scripts for running QTCM on CRAY is also provided.

2.0 Released August 1998. Major changes and additions from version 1.0:

- Method for calculating wind speed used in the evaporation parameterization is altered.
- Dynamical equations now account for spherical geometry. See Section 2.3.3 for more details.

- `noout` run-time option added.

1.0 Released July 1997. This version is the β-version of the model. At the time of release, this version was numbered 2.0. In Summer 1998, it was retroactively renumbered 1.0; in the source code for v2.0 and later versions, this version is referred to as v1b. This was the version used in the Lin et al. [6] paper on the Madden-Julian Oscillation.

original January 1997. Not publicly released. In the source code for v2.0 and later versions, this version is referred to as v1a. This is also referred to as the α-version of the model.

1.1.3 Model Spatial and Temporal Coverage

The standard version of the QTCM1 provides coverage from 78.75°S to 78.75°N in latitude and coverage over all longitudes. The grid has dimensions $nx \times ny$ equal to 64×42. The east/west boundaries are periodic, and the north/south boundaries are solid walls. Thus, the grid spacing is 5.625 [i.e. $360/nx$] degrees longitude and 3.75 [i.e. $2 \times 78.75/ny$] degrees latitude.

The first index increments in the zonal (x) direction with increasing longitude in the eastward direction, beginning at 0 degrees E longitude. The second index increments in the meridional (y) direction with increasing latitude northward, beginning at 78.75°S latitude (i.e. $-78.75°$ latitude). Thus:

grid box (1,1) is centered at 0°E longitude, $-76.875°$ latitude
grid box (2,1) is centered at 5.625°E longitude, $-76.875°$ latitude
grid box (2,2) is centered at 5.625°E longitude, $-73.125°$ latitude
grid box (nx, ny) is centered at 354.375°E longitude, 76.875° latitude

The spatial array sizes nx and ny are set in the module file qtcmmod.f90 Higher resolution can be easily achieved by modifying hgrid.h and subroutine `parinit`. Higher resolution boundary data sets can be created using a package released along with this version of model. See Section 1.4.3 for details. Section 2.2.3 has a more complete description of the model grid.

1.1.4 Miscellaneous Notes

Unless otherwise described, all variables represent the "full" value of the field. Thus, "zonal velocity" is the "full" (mean plus anomaly) zonal velocity, not just the anomaly or mean.

Names of quantities or parameters (such as zonal velocity u), are set in an italicized font.[1] Names of model variables (e.g. barotropic zonal velocity `u0(i,j)`), which may or may not have a corresponding similarly titled physical quantity, are set in a non-proportional "typewriter" font.[2] File names, directories, shell scripts, program commands, and subroutine and function names are also generally set in a non-proportional verbatim font.

A representation of topography is in the code but is turned off in the standard version.

1.2 What's Included In the Model

The tar package for this release of the QTCM1 model contains the following subdirectories:

bnddata Directory of boundary data files, such as sea surface temperature (SST), albedo, etc. See Section 1.4.1 for more information.

doc The LaTeX files for this manual, as well as an HTML version and a Postscript version.

inidata Initial conditions (restart files).

misc Documentation and possibly notes and errata.

proc GrADS scripts that can be used to display and analyze the output from the model.

src Source code for the model.

work This directory provides a workspace in which to execute the model. Section 1.3.2 has a more detailed description of this directory.

Because the model is meant to be reasonably straightforward to diagnose, the complete model consists of comparatively few files. Below is a listing and short description of the source code/parameter files that make up the model; they are found in directory `src`:

[1] The LaTeX "math mode" environment, "italicized text" command, or the TeX "italicize" environment is used to create this formatting.

[2] The LaTeX "verbatim" environment or typewriter family text-producing command is used to create this formatting.

abl.f90 Simple atmospheric boundary layer model for surface winds and to compute surface fluxes.

bndry.f90 A generic routine for reading in monthly climatological fields.

calendar.f90 Source code for calendar and time managing routines.

clrad.f90 Source code for radiation parameterization package by Chia Chou [5]. The solar component is diurnally averaged.

clrad0.f90 Source code for a simpler radiation parameterization package. The solar component is diurnally averaged.

clrad1d.f90 Source code for radiation parameterization package by Chia Chou [5], with the diurnal cycle resolved.

cplmean.f90 Source code for routine that accumulates coupling variables (fluxes) from the `qtcm` model over time for ocean coupling.

driver.f90 Source code for subroutine `driver` which couples the atmospheric model and the SST boundary condition which drives it.

fatdpkg.f90 Direct Poisson solver, courtesy of Alistair Adcroft.

getbnd.f90 A routine that passes boundary data arrays such as SST and albedo.

land.f90 The land-surface model Simple-Land (SLand1); and a simple implementation of the bucket model (not used).

makefile The makefile to compile the model.

ocean.f90 Source code for routine that provide QTCM with SST ("ocean model") Per default this is a routine which reads SST from a data file. If `MXLOCEAN` is define a slab mixed layer ocean is used.

output.f90 Source code for output routines.

par.f90 Source code for a program "par" which computes the model parameters from the sounding table in file Tqo3.ref.

qtcm.f90 Source code for routines in the QTCM1 atmospheric model. These subroutines are called and controlled by subroutine `qtcm`.

qtcmmod.f90 QTCM module file. The version 2.2 common block definition in header files hgrid.h qtcm.h, driver.h and calendar.h are merged in to modules stored in this file.

qtcmpar.f90 Parameter module file. This file is generated by the program "par" whose sources are in `par.f90`.

sflux.f90 Source code for surface flux computation.

Tqo3.ref Reference profile of temperature, humidity and ozone used as input of program "par" to compute the model parameters.

utilities.f90 Source code for various utility routines.

All source code is written in FORTRAN-90.

1.3 How to Make a Basic Model Run

1.3.1 Compiling the Model

To test if the model compiles on your machine, select a suitable makefile in the `src/Makefiles` directory and copy it into the `src` directory under the name `makefile`. Then go into the source directory `src` and execute "make". If things fail you will have to adjust the makefile to your system. The different makefiles in `src/Makefiles` provide you with some guidance on how to change it on different systems. We successfully compiled QTCM on Compaq Alpha (DEC) OFS1, Cray J90, IBM AIX, PC Linux (Intel ifc and Lahey-Fujitsu compiler), Sun Sparc and SGI Irix machines for which we provide example makefiles in the `src/Makefiles` directory. If you run into problems please let us know. Due to the various preprocessor macros in the code the compile process has to include a preprocessor step. Have a look at your Fortran-90 documentation to see how to activate it.

N.B.: Unfortunately, some Fortran-90 insist on specific file name extensions to correctly compile Fortran-90 code with preprocessor directives. One group of compilers prefers an ".F90" extension to include a preprocessor step while others refuse to compile anything with such an extension. We chose the ".f90" extension which works on most systems. Sometimes one has to specify a compiler specific flag to enable the Fortran-90 preprocessor (see `FPPON` in the makefiles). The exceptions are the Lahey-Fujitsu (PC-Linux) and the IBM-AIX compilers for which the file name extension hasto be changed. The makefiles for these compilers do this automatically the first time the model is compiled.

1.3.2 Running the Model

To run the model, go to the workspace directory `work`. This directory contains scripts to execute the model.

The package comes with a number of different C-shell scripts that control different ways of model execution. In each script you can define macros for specific physics (see section 1.3.3). These scripts are:

spinup.sh Makes the "spin-up" run of QTCM1. This creates a qtcm.restart file that can be used to shorten the "spin-up" of the model. This script uses monthly climatological SSTs as forcing.

realtime.sh Runs QTCM1 using SSTs interpolated from observed values. The default SST dataset is from November 1981 to October 2001. This can be reset for any period if SST data is supplied.

seasonal.sh Runs QTCM1 using monthly climatological SSTs.

perpetual.sh Runs QTCM1 using perpetual SSTs.

mxlayer.sh Runs QTCM1 coupled with the mixed-layer ocean. Macro "MXL_OCEAN" is turned on. It typically includes three steps. First, QTCM1 is run with observed SSTs. Second, annually or monthly- averaged Q-flux is computed using aveflux.f. Lastly, QTCM1 is run with the mixed-layer ocean. Always use YEAR360 calendar as assumed by Q-flux calculation program aveflux.f.

blendsst.sh Runs the model using prescribed observed SST in masked regions and mixed-layer ocean in non-masked regions. Macro "BLEND_SST" is turned on. Use YAER360 calendar, same as above.

ensemble.sh Script for a 10 member ensemble run. The 10 initial condition files can be generated by a 10 year model run with seasonal SST using the script ensemini.sh.

ensemini.sh Creates 10 initial condition files for ensemble simulations. Macro "ENSEMB_INI" is turned on.

test.sh Executes a brief run of QTCM1 to test code-machine compatibility.

To run the model, change the user-controlled parameters in one of the above scripts to suit your purposes, then execute the scripts by typing the script name at the command line. After you first port the model over to your computer, you should run `test.sh` to confirm that the model will compile and run correctly on your machine. Next, run the model a year or so with climatological SST to create a qtcm.restart initial conditions file that will serve to shorten the spin-up time of the model. The `spinup.sh` script is set-up to do this. Finally, conduct an actual model run by executing `qtcm_realtime.sh` or `qtcm_seasonal.sh`, as appropriate.

The model execution scripts are all very similar. They go through the following steps:

1. Definition basic script parameters (preprocessor switches and directories).
2. Compilation of qtcm in the `src` directory.
3. Creation of an output sub-directory.
4. Documentation of the run by creation of a compressed archive in the output directory which contains the source code and the run script.
5. Collection of all files to run the model in the output directory (executable, input parameter and restart file)
6. Model execution in the output directory, saving the script and the model output in a log file and displaying the model's output diagnostic on screen.

1.3.3 Macros

The following are the Macros available for defining spectific physics of the model. They are found in the makefile in the verb—src— directory.

CPLMEAN Compute mean flux values for coupling with an ocean.

LINEAR_T1C Use the linear moisture closure.

YEAR360 A year has 360 instead of the 365 days (no leap years).

NETCDFOUT Write output in netCDF instead of Grads sequential binary

SPONGES Activate sponge code (parameters have to be set in the code).

DELTA_CO2 Activate radiation code for changed CO2 level (default: 330 ppm).

NO_ABL Old version 2.2 surface fluxes: Estimates without Atm. Boundary Layer

MXL_OCEAN Use mixed layer ocean

BLEND_SST Use readsst in mask regions and mixed layer ocean outside.

TOPO Turn on topography effects due to induced divergence.

CHIWIND Diagnose velocity potential using irrotational winds.

ENSEMB_INI Create initial condition files for ensemble runs.

1.3.4 Model Input Parameters

Below is an explanation of the user-controlled variables in the models input file `driver.in`. Default values for these user-controlled variables are defined in the subroutine `DriverInit` in `driver.f90`:

bnddir Path and name of the directory containing the boundary condition datasets. See Section 1.4.1 for details.

SSTdir Path and name of the directory containing the SST datasets. See Section 1.4.1 for details.

outdir Path and name of the directory to which the output data will be written. See Section 1.4.2 for details.

runname Short string used in the output filename.

landon Land-interactive indicator: `landon=1` (the default) means land is interactive. Other values of `landon` mean the land is treated like ocean with a fake SST provided from the SST files. Since an interactive land has more freedom, it is more prone to numerical instability in the spin-up process, and it has spin-up time longer than the atmosphere by several months. You may never encounter this problem. Our recommendation is to run the model with land “off” for the first time (`spinup.sh` is set with such a setting), or use the default file qtcm.restart.

SSTmode Model running mode: Affects how SST is fetched and how `solartop` evolves. Variable is an alphanumeric string. There are three mode options:

- `perpetual`: Perpetual SST from a given month is used (but watch out for summer continents that may be too warm if the land is "on"). Month used is specified in `month0`. When the mixed-layer ocean model is active, this mode means annually averaged Q-flux (see Section 3.4) is used.
- `seasonal`: Climatological monthly SST; solar top seasonal. When the mixed-layer ocean model is active, this mode means seasonal Q-flux is used.
- `real_time`: Real time interannual SST; solar top seasonal.

year0, month0, day0 Starting date for the model (note `year0` includes the millennium and century). For `perpetual` and `seasonal` cases, `year0` can be any value between 1 and 9998. However, note that year0 should be chosen such that during the *entire* length of the model run, the model year never exceeds 9998. Type INTEGER. Note: 1 year $\equiv$ 365 days (no leap years) in QTCM1 for standard setting. A calendar of 360 days a year can be switched on by defining the macro YEAR360 in the makefile.

lastday Model runs from day 1 to `lastday`. Type INTEGER.

interval Ocean-atmosphere coupling interval (days). Must be greater than or equal to 1. Type INTEGER.

noout No output for the first `noout` days; useful for spin-up. Type INTEGER.

ntout Mean output every `ntout` days. See Section 1.4.2 for more details. Type INTEGER. For the standard 365 day calendar `ntout`$= -30$ specifies monthly output. `ntout`$=0$ suppresses the output of averages and disables the averaging routine, which make the model about 10% faster.

ntouti Instantaneous output every `ntouti` days. See Section 1.4.2 for more details. Type INTEGER. `ntout`$= -30$ specifies monthly output. `ntout`$=0$ suppresses the any output of instantaneous fields.

ntoutr Restart file output every `ntoutr` days. `ntout`$= -30$ monthly restart file. `ntout`$=0$ no restart file except at end of model run.

mrestart Flag controlling whether to use qtcm.restart. If `mrestart=1` (the default), model uses file qtcm.restart as initial condition (it is recommended to always use this option, except for the first time when you need to generate your own qtcm.restart file, together with the land off). Otherwise, the model will start from an unbalanced initial state; the spin-up time for the atmosphere is several days, and several months for land. Type INTEGER.

NB: Currently, restart using qtcm.restart is not "perfect." This is because in the standard version, qtcm.restart.out does not save the atmospheric mixed layer surface winds which are use as the initial for the new winds.

dt Time step for the atmosphere (sec). Type REAL.

mt0 Dimensionless multiplier to set barotropic mode time step. The time step (sec) for the barotropic mode equals `dt` × `mt0`. Default value is 1. Type INTEGER.

viscT Temperature diffusion coefficient $[m^2/s]$.

viscQ moisture diffusion coefficient $[m^2/s]$.

visc4U Forth order velocity diffusion (*"hyper-viscosity"*) coefficient $[m^2/s]$.

ziml Atmospheric mixed layer depth $[m]$.

weml Atmospheric mixed layer entrainment velocity [m].

Wsmin Minumum wind speed for surface flux calculation [m/s].

V1b Mode 1 velocity projection coefficient at the top of mixed layer [m/s].

arr1name, . . ., arr8name Auxiliary array names for output. There are 8 auxiliary arrays named `arr1`, . . .,`arr8`. These arrays are include in the output if the cor responding arrXname is set. The structure of the arrXname string has to be "variable name", one or more spaces,

Below is a sample of the user-controlled parameters section from a model execution script (e.g. `test.sh`):

```
&driverdata
title='A QTCM test run'
bnddir='$BNDDIR'
SSTdir='$BNDDIR/SST_Reynolds'
outdir='$OUTDIR'
runname='$RUNNAME'
landon=0
lastday = 61
ntouti=-30                          ! monthly instantaneous data output
mrestart=0
!
! default values set in driverInit:
! ---------------------------------
```

```
!  title='QTCM default title'          ! A descriptive title
!  bnddir='../bnddata'                 ! bnd. data other than SST
!  SSTdir='../bnddata/SST_Reynolds' ! where SST files are
!  outdir='../proc/qtcm_output'        ! where output go
!  runname='runname'                   ! string used to create an output filename
!  landon=1                            ! if not 1: land is like ocean with fake 'SST'
!  SSTmode='seasonal'                  ! decide what kind of SST to use
!  year0 = -1                          ! starting year if <0 use year in restart file
!  month0 = -1                         ! starting month if <0 use month in restart file
!  day0 = -1                           ! -1 and mrestart=1: Use date saved in the restart file
!  lastday = 365 !(for YEAR360: 360)   ! last day of integration
!  interval = 1                        ! coupling interval
!  noout=0                             ! no output for the first noout days
!  ntout=-30                           ! monthly mean output
!  ntouti=0                            ! monthly instantaneous data output
!  ntoutr=0                            ! restart file only at end of model simulation
!  nooutr=0                            ! Days to skip before writing restart files.
!  mrestart=1                          ! =1: restart using qtcm.restart
!  dt=1200.                            ! time step [seconds]
!  mt0=1                               ! barotropic timestep every mt0 timesteps
!  viscT=12.0e5                        ! temperature diffusion parameter [m^s/s]
!  viscQ=12.0e5                        ! moisture diffusion parameter [m^s/s]
!  viscU=7.0e5                         ! viscocity parameter [m^s/s]
!  ziml=400                            ! Atmos. Mixed layer depth [m]
!  weml=0.02                           ! Mixed layer entrainment velocity [m]
!  Wsmin=4.5                          ! Minumum wind speed for flux calculation [m/s]
!  V1b=-0.2204077                      ! V1 projection coeffcicient at top of mixed layer
!  arr1name='?'                        ! Auxiliary output array names 1...8
!  arr2name='?'                        !='?' Array not included in output.
!  arr3name='?'
!  arr4name='?'
!  arr5name='?'
!  arr6name='?'
!  arr7name='?'
!  arr8name='?'
! Example:
! arr1name='dps1dx Mode 1 contribution to dpsdx [m/s^2]'
!-----------------------------------------------------------------
&end
```

This sample defines a run using monthly climatological SST as forcing ending after 59 days. Land acts like an ocean with interpolated SST, no “spin-up” file is used for initial conditions.

1.3.5 Processing and Viewing the Output

The standard QTCM1 package includes GrADS scripts, to compute various quantities from the output files. These files are found in directory `proc`. These are intended to provide helpful examples of common diagnostics for users to modify! (Please note that we don't have the personnel to support all potential variants of diagnostics).

The GrADS scripts provided do not need to be compiled; just run them within the

GrADS program. For more information on the GrADS system go to http://grads.iges.org/grads. The output files from QTCM1 provide GrADS binary (Fortran sequential unformatted) data. Some of the GrADS scripts and control files provided with with the QTCM1 tar package include:

ca.gs Generic routine to calculate climatology and anomaly for arbitrary periods.

clim.gs Four season plots for precipitation, fluxes and winds.

ensocomp.gs Script to plot composite ENSO precipitation anomaly in DJF and JJA.

flux.gs Calculates various radiative fluxes.

nino3corr.gs Script to plot maps of Nino3-Precipitation and Nino3-surface temperature correlations.

nino3.gs Script to calculate various statistical indices associated with NINO3.

qtcmpar.gs Script for defining QTCM1 parameters used in calculating surface wind, etc. Automatically generated by par.f90.

timelongEQ.gs Script for time-longitude plot of precipitation on Equator.

u.gs Calculates winds at different model levels.

vert.gs Script to plot the vertical cross section of climatological zonally-averaged winds.

All the scripts listed above require that you first open a GrADS control file before use. Sample figures for this and other versions of the model may be found at `http://www.atmos.ucla.edu/~csi/QTCM/sampfigs.html`. For more information on visualizing results using other graphics packages see Section 1.4.2.

1.4 Descriptions of Files and Scripts

1.4.1 Input Files

Boundary Condition Files

These files need to be in the directory whose path and name are given by the execution shell script variable `bnddir` in order for the model to work. In this release, this directory is `bnddata`.

All files are ASCII formatted vertical lists arranged using standard FORTRAN looping conventions for an array, which first increments the first index, then the second index, and so on. Thus, each line the boundary condition file corresponds to the value of the variable in a spatial domain grid box in the following way:

line 1 corresponds to grid (1,1)
line 2 corresponds to grid (2,1)
line nx corresponds to grid (nx,1)
line $nx + 1$ corresponds to grid (1,2)

The following datasets are included in the boundary condition directory:

ALBD_Darnell Climatological monthly observed surface albedo derived from Darnell et al. [21].

CLOUD_ISCCP Climatological monthly ISCPP cloud cover. This cloud data is used in the radiation scheme by Chou and Neelin [5], which is implemented in clrad1.f90. for model initialization.

SST_Reynolds Directory containing Reynolds [29] blended SST. See below for more information.

STYPE Surface type: ocean = 0, forest = 1, grass = 2, desert = 3. Note that the array `STYPE` is declared REAL∗8 in the code.

TOP Smoothed relative topography; height/10 km (currently the same as the file TOP.smooth).

A package of Fortran programs used to generate boundary data at arbitrary model resolution and domain size is provided. It can be downloaded from the web site

`http://www.atmos.ucla.edu/~csi/QTCM/`

Detailed description of the package can be found in section 1.4.3.

Sea Surface Temperature (SST) Files

These files are taken from Reynolds' [29] blended SST data files. Each month's data is in an individual file. These files also follow the format convention described in Section 1.4.1.

The SST datasets are found in the directory whose path and name are given by the execution shell script variable `SSTdir`. In this release, the default location of these files is `SST_Reynolds`.

Climatological SST have filenames of the form:

0000mmdd.sst

Real-time SST have filenames of the form:

yyyymmdd.sst

In both cases, "mm" is month (e.g. 02, 10), "dd" is day, and "yyyy" is year (e.g. 0000, 1991). For all SST files, we assume the month's data is centered on the 15th, so "dd" equals "15". For the real-time SST, the supplied data range from November 1981 to October 2001. Monthly data are linearly interpolated to daily averages in rdsst.f.

Model Physics Parameters Files

The file qtcmpar.f90 (in `src`) contains the value of various physical parameters used by the model. The program that generates many of these parameters, par.f90, is provided but is not easy to use, since some parameters have subtle physical and numerical properties. We don't recommend modifying most parameters unless you are quite familiar with the model.

Below is a brief description of some of the categories of variables described in qtcmpar.f90:

1. Variables `a1hat`, `a1s`, `V1s`, `b1hat`, `bb1hat` and `b1s` help describe the temperature and humidity profiles.

2. Variables `GMsr`, `GMsp`, `GMqr`, `GMqp`, `GMs0r`, and `GMq0r` help describe the gross moist stability.

3. `eps_c` is used in our version of the Betts-Miller [17, 18, 19] convective parameterization.

4. `CV0` is related to the neutral drag coefficient. See subroutine `sflux` in qtcm.f90 for more details.

5. Variables `eps_0`, `eps_10`, `eps_01`, and `eps_1` are related to momentum damping. These parameters are obsolete.

6. `Tsref` is the reference surface temperature.

7. `Cpg` converts heat flux into column heating rate.

8. Parameters `Trefs`, `qrefs`, `Tcrefs`, `qcrefs`, `Trefhat`, `qrefhat`, `Tcrefhat`, and `qcrefhat` are related to the reference atmospheric profile.

9. Parameters `Vijk`, `Vwijk`, `DTijk`, and `Dqijk` are coefficients for advection of momentum, temperature and moisture.

The figure below gives a reproduction of the file. Chapter 3 gives a fuller discussion of the physics behind the variables in this file. Section 2.4.2 relates model variables with the variables given in the QTCM1 main papers [1, 2].

```
Module Qtcmpar
  !
  ! QTCM mode parameter, generated by par.F90
  !
  Integer, Parameter  ::  &
       &       nvmod  =   2  & !  no. v-modes
       &     , nTmod  =   1  & !  no. T-modes
       &     , nz     =  10    !  vertical z-resolution for topography
  REAL, Parameter     ::  &
       &       a1hat    = 0.45934841      & ! hat{a_1}   [-] NZ (3.2)
       &     , a1phat   = 0.24520899      & ! hat{a_1^+} [-] NZ (2.16), (3.9)
       &     , a1phatb  = 0.15463794E-01 & ! average of a_1^+ over subcloud layer [-]
       &     , a1s      = 0.30203986      & ! a_{1s})      [-] NZ (5.16)
       &     , V1s      = -.24520899      & ! V_{1s}       [-] NZ (4.13)
       &     , V1sqhat  = 0.39553840E-01  & ! hat{V_1^2} [-] NZ (4.13)
       &     , b1hat    = 0.31574178      & ! hat{b_1}   [-] NZ (5.17)
       &     , b1s      =  1.0000000      & ! b_{1s}       [-] NZ (4.32)
       &     , bb1hat   = 0.37340307      & ! hat{B_1}   [-] NZ (4.26)
       &     , GMsr     =  3.5000000      & !  M_{sr}      [K]
       &     , GMsp     = 0.42756125E-01  & !  M_{sp}      [-]
       &     , GMqr     =  3.0000000      & !  M_{qr}      [K]
       &     , GMqp     = 0.50721642E-01  & !  M_{qp}      [-]
       &     , GMs0r    =  44.398193      & !              [K]
       &     , GMq0r    = -51.881908      & !              [K]
       &     , eps_c    = 0.13888889E-03  & !  [1/s]
       &     , CV0      = 0.14000000E-01  & !  reference drag coeff*velocity [m/s]
       &     , eps_0    = 0.19369413E-05  & ! [1/s]=>  6.0    [day]
       &     , eps_10   = -.47495544E-06  & ! [1/s]=> -24.    [day]
       &     , eps_01   = -.12007821E-04  & ! [1/s]=>-0.96    [day]
       &     , eps_1    = 0.29444259E-05  & ! [1/s]=>  3.9    [day]
       &     , eps_i1   = 0.89764910E-06  & ! [1/s]=>  13.    [day]
       &     , Tsref    =  302.26001      & !  surface reference temperature  for radiation [K]
       &     , Cpg      =  8708163.0      & !  conversion factor Cp*delp/gravity [J/K/m*2]
       &     , Trefs    =  302.00000      & !  surface reference temperature [K]
       &     , qrefs    =  51.955292      & !  surface reference humidity    [K]
       &     , Tcrefs   =  302.00000      & !  convective surface ref. temperature [K]
       &     , qcrefs   =  50.814919      & !  convective surface ref. humidity [K]
       &     , Trefhat  =  267.77045      & !  mean reference temperature [K]
```

```
     &     , qrefhat  =  16.404453      & !  mean reference humidity [K]
     &     , Tcrefhat =  268.98325      & !  [K]
     &     , qcrefhat =  16.159267        !  [K]
 !
 ! advection coefficients
 !
 REAL, Parameter, Dimension(nvmod**3) ::  &
     &    Vijkt = (/   1.0000000    , 0.0000000       & ! Vijk(:,0,0)
     &              ,  0.0000000    ,0.39553840E-01   & ! Vijk(:,1,0)
     &              ,  0.0000000    , 1.0000000       & ! Vijk(:,0,1)
     &              ,  1.0000000    ,0.11134213    /) & ! Vijk(:,1,1)
     &  , Vwijkt = (/   0.0000000    , 0.0000000      & ! Vwijk(:,0,0)
     &             , 0.24520901    ,0.39485272E-01    & ! Vwijk(:,1,0)
     &             ,  0.0000000    , 0.0000000        & ! Vwijk(:,0,1)
     &             , -.26275271    ,0.55228624E-01/)    ! Vwijk(:,1,1)

 REAL, parameter, dimension(nvmod*nTmod**2) ::  &
     &    DTijkt = (/   1.0000000    ,0.68461813E-01  /)  & ! DTijk(:,1,1)
     &  , Dqijkt = (/   1.0000000    ,-.16064279      /)    ! Dqijk(:,1,1)

 REAL, Parameter, Dimension(nz) ::  &
     & V1z = (/ -.24520899    , -.21125591    , -.17568490    , -.13640948    , -.92673302E-01   &
     & , -.43544598E-01, 0.12078471E-01, 0.75964749E-01, 0.14977413    , 0.23455063     /)
 REAL, Parameter, Dimension(5) ::   & ! a1p  at 1000, 950, 900, 850, 800 mb
     & a1p = (/  0.0000000    , 0.15051006E-01, 0.30990627E-01, 0.47977556E-01, 0.66651829E-01 /)
End Module Qtcmpar

Module T1cTableIn
 ! Profile for nonlinear T1c lookup table
 Integer, Parameter :: np=         11
 REAL, Parameter, Dimension(4*np) ::  table= (/&
    !          p [HPa]  ,     alpha [-]   ,    Tsat [K]  ,     a1 [-]
     &     1000.0000    , 0.85000002    , 302.00000    ,0.30203986    ,   &
     &     950.00000    , 0.83529413    , 297.64838    ,0.29764515    ,   &
     &     900.00000    , 0.82058823    , 293.08527    ,0.29308209    ,   &
     &     850.00000    , 0.80588239    , 291.01846    ,0.30323407    ,   &
     &     800.00000    , 0.79117650    , 288.79773    ,0.31495300    ,   &
     &     700.00000    , 0.76176476    , 283.78235    ,0.34451482    ,   &
     &     600.00000    , 0.73235297    , 277.72491    ,0.38630152    ,   &
     &     500.00000    , 0.70294118    , 270.06317    ,0.44864288    ,   &
     &     400.00000    , 0.67352945    , 259.69208    ,0.54513776    ,   &
     &     300.00000    , 0.64411765    , 244.30972    ,0.67801291    ,   &
     &     200.00000    , 0.61470592    , 219.90631    ,0.74564534        /)
End Module T1cTableIn
```

1.4.2 Output Files

All output is written to the directory `proc/outdir` specified in the run script. The model can generates two output files in either netCDF format or sequentially unformatted binaries. The latter is the default. NetCDF output can be requested by defining the preprocessor macro `NETCDFOUT` in the run script or the makefile (See Section 1.3.3.

qm_name.ext Average values of all the prognostic variables and major diagnostic variables, where `name` is replaced by the valuer of the input string `runname` or build from the current date and time if `runname` is not specified. The file extension `.ext` is ".out" for Fortran sequential binary output and ".nc" for output in netCDF format. The data are output every `ntouti` days after skipping the initial `noout` days.

qi_name.ext Same as above for instantaneous values.

qtcm_YYYYMMDD.restart QTCM restart file at end of model date `YYYY`,`MM,` verb—DD—, where `YYYY`, `MM` and `DD` are model year, month and day, respectively. The restart file is written every `ntoutr` days after skipping the initial `nooutr` days. For `ntout`, `ntouti` and `ntoutr` the special values of -30 and 0 indicates monthly and no output, respectively.

Control of Output

As noted above, the frequency of mean and instantaneous output is controlled by the variables `noout`, `ntout` and `ntouti` in the input namelist `driverdata` in the run script. No instantaneous or mean output can be specified by setting `ntouti=0` or `ntout=0`, respectively.

The user can make additional adjustments as to what is output by making code changes to output.f90 found in `/src`. Some of the adjustments possible include:

- To skip an output variable comment out the corresponding call of `defVar` in the subroutine `defOutVars` located near the top of the source file `output.f90`.

- To add an array variable for output add an appropriate call of `defVar` to the subroutine `defOutVars` in `output.f90`. The subroutine syntax is `defVar(longvarname,var)`, where `var` is the variable to be included in the output and `longvarname` is a string describing it. The first word of `longvarname` is split off and used as the variable name in the netCDF file or GrADS data descriptor file, respectively. For netCDF output the first substring of `longvarname` in brackets is used as the "units" attribute to the output variable.

  Example:

  ```
  Call defVar('diffT1 T1 diffusion  [K/s]',dfsT1)
  ```

requests to include the fortran array named `dfsT1`. It's name will be `diffT1`, the description string is "`T1 diffusion  [K/s]`" and, for netCDF, the units are "`K/s`":

The output routines automatically allocate storage for the mean computation. No array size adjustments are necessary. The output routines use pointers and a linked list to access the array data so the array has to be declared with a Fortran-90 "`TARGET`" attribute. This means that you may have to add the `TARGET` keyword in the variable declaration (usually in `/src/qtcmmod.f90`). Furthermore, the array has to be in the scope of the `defOutVars` routine so you may have to add it in the list of used modules in `defOutVars`. You may use the already existing variables as a guidance. Usually only `defOutVars` needs to be adjusted. The calling sequence is slightly different for netCDF and GrADS binary output (see section 2.2.1).

By default, output is written every `ntout` days to the three files described above. However, the code allows the user to specify a different output frequency and unit directly for any particular variable in subroutine `outpinst` or `outpmean` by simply using different argument values in calling subroutines `outpi` and `outpm`. Variables `ntout` and/or `ntouti` can also be specified as a user-control parameter in driver.in by editing the execution shell scripts. Note: The minimum value for `ntout` is one day, so if you need to output more often than a day, you have to call the output routines from within the atmosphere-land coupling loop in subroutine `qtcm`.

Content of Output

Binary Output The default output format is sequential unformatted binary output. The binary format is the native format of the machine. The variables are written so that the data can be easily displayed and analyzed with GrADS (see `http://grads.iges.org/grads`). The variables are written in the following manner:

- The time steps are written out in ascending order.
- At each time step, the entire spatial grid for each variable is output to the file.
- Each spatial grid is written out in conventional FORTRAN array format (with the first index written out in ascending order, then the second index, and so on). Thus, the array u_1 for zonal wind (baroclinic mode) is written out in the following order:

$$
\begin{array}{l}
u_1(1,1), u_1(2,1), u_1(3,1), \ \ldots \ , u_1(nx,1) \\
u_1(1,2), u_1(2,2), u_1(3,2), \ \ldots \ , u_1(nx,2) \\
u_1(1,3), u_1(2,3), u_1(3,3), \ \ldots \ , u_1(nx,3) \\
\qquad\quad \vdots \\
u_1(1,ny), u_1(2,ny), u_1(3,ny), \ \ldots \ , u_1(nx,ny)
\end{array}
$$

Note that for the meridionally staggered variables that are dimensioned in the y-direction from 0:ny, the southern boundary `(i,0)` is not output. This is because the value at the southern boundary is set to zero in the model. See Section 2.3.1 for more details.

- When meridionally staggered variables are output, QTCM1 simply writes the array. Grid interpretation is written for the location of the `T` grid points. QTCM1 does not interpolate staggered variables to the `T` grid points (i, j). Thus, if you wish to have the "true" value of the meridionally staggered variables at (i, j), you will need to interpolate accordingly or supply offset information in the plotting routine. The plotting routines described in section (1.3.2) actually ignore grid offset in plotting u,v. Some of these meridionally staggered variables include `u1`, `v1`, `u0`, and `v0`. See Section 2.2.3 for more details.

- The order of variable output is that given in the generated control file.

Note that some of the variable units given in Table 1.1 are "non-conventional," i.e. they are in units different from those in which they are usually found. For instance, precipitation is usually given in mm/day; the model outputs the parameter in W/m^2. A few conversion factors to more "conventional" units are given below (see the GrADS script `qtcmpar.gs` for more conversion factors):

TO CONVERT	FROM	TO	ACTION
`Evap`	W/m^2	mm/day	Divide W/m^2 by 28.2
`q1`	K	g/kg	Divide by 2.42
`Prec`	W/m^2	mm/day	Divide W/m^2 by 28.2

Note also that because output files are written in sequential access, they have FORTRAN control words embedded in the records. Thus, in order to properly read the output files, the `sequential` option needs to be specified in GrADS, and the `/F77_UNFORMATTED` keyword needs to be used in the IDL `OPEN` procedure.

VARIABLE	DESCRIPTION	UNITS
`u1`	Zonal wind, baroclinic mode	m/s
`v1`	Meridional wind	m/s
`T1`	Air temperature	K
`q1`	Humidity	K
`u0`	Zonal wind, barotropic mode	m/s
`v0`	Meridional wind	m/s
`vort0`	Barotropic vorticity	s^{-1}
`psi0`	Barotropic streamfunction	m^2/s
`Ts`	Surface temperature: for ocean is SST, for land is ground temperature	K
`Prec`	Precipitation (a.k.a. Q_c)	W/m^2
`cl1`	Cloud amount[a]	fraction
`S0`	Incoming solar radiation at TOA[b]	W/m^2
`FSWds`	Downward shortwave flux at surface	W/m^2
`FSWus`	Upward shortwave flux at surface	W/m^2
`FSWut`	Upward shortwave flux at TOA	W/m^2
`FLWds`	Downward longwave flux at surface	W/m^2
`FLWus`	Upward longwave flux at surface	W/m^2
`OLR`	FLWut in code, upward longwave flux at TOA	W/m^2
`FTs`	Sensible heat flux	W/m^2
`Evap`	Evaporation	W/m^2
`Runf`	Runoff	W/m^2
`WD`	Equivalent water depth[c]	kg/m^2
`STYPE`	Surface type (ocean = 0, land > 0)	
`taux`	Surface stress along x	N/m^2
`tauy`	Surface stress along y	N/m^2
`advT1`	Negative of advection projected onto `T1`	m/s^2
`advq1`	Negative of advection projected onto `q1`	m/s^2
`Evapi`	Interception loss, in W/m^2	Evapi
`GMq1`	Moist stratification `[K]`	M_q
`wet`	Relative wetness	W/W_0
`Runs`	Surface runoff, in `[W/m^2]`	R_s

[a]In clrad0.f90, this variable represents high and middle clouds (including deep convective clouds). In clrad1.f90 and clrad1-d.f90, this variable is deep convective cloud fraction.

[b]TOA = "top of atmosphere".

[c]1 kg/m^2 equals 1 mm equivalent depth for the case of liquid water, if we assume a density of 1000 kg/m^3.

Table 1.1: Standard output variables.

GrADS descriptor file Besides the binary file a suitable GrADS data descriptor file is automatically generated in the same directory as the binary data file. It's file name is the binary file with the extension `.out` replaced by `.ctl`.

NetCDF Output The output files contains the same data as the GrADS binary file plus some information about used the model input parameters. The files obey the COARDS convention (see `http://ferret.wrc.noaa.gov/noaa\_coop/coop\_cdf\_profile.html`) and may be opened using the command "sdfopen" within GrADS. For more information on "sdfopen" see `http://www.cdc.noaa.gov/~hoop/grads.html`.

1.4.3 Other Scripts and Files

aveflux.f Fortran program used to calculate Q-flux from a control run. **N.B.:** This program assumes that the output is written in binary format and that no output data was added or eliminated.

bnddataV2.1.tar.gz Gzipped and archived Fortran programs to generate boundary data (surface type, topography, albedo and Sets) at arbitrary model resolution and domain size. When gunzipped, the file structure is similar to /bnddata in QTCM1 with the addition of two sub-directories `raw` and `bin`. `raw` stores all raw data. `bin` stores shell scripts and Fortran programs for interpolation. To execute, run README file in `bin`. Dimension parameters are specified in README.

1.5 Overview of Model Flow

Figure 1.1 briefly shows the interconnections between the different routines used in the model. A more detailed description of the routines and the code, as well as a complete calling tree (Section 2.2.1) is found in Chapter 2.

```
Driver ===> DriveInit
            OceanInit
            qtcminit  ===> parinit
                           bndinit
                           varinit
                           physics1
            outpinit
            Do (ocean-atmosphere coupling)
              Ocean
              QTCM    ===> getbnd ===> getSST
                                       bndry1
                           Do (land-atmosphere coupling)
                             physics1 ===> mconvct
                                           cloud
                                           radsw
                                           radlw
                                           sflux
                             land
                             atmostep ===> advctuv
                                           advctTq
                                           dffus
                                           barcl
                                           bartr
                             varmean
                           Enddo
              outpall ===>  outpinst
                            outpmean
            Enddo
            finish
```

Figure 1.1: Overview of model flow

Chapter 2

Model Code

2.1 Introduction and Overview

The QTCM1 prognostic equations are finite-differenced on an Arakawa C-grid. The barotropic component is solved in a vorticity/streamfunction formulation using an Adams-Bashforth scheme. The baroclinic component is similar to a shallow-water equation and is solved by applying a "forward-backward" scheme using the updated variables instantly. This forward-backward scheme is described by Haltiner and Williams [24][1]. The numerical CFL instability criterion limits the model time-step at about 20 minutes, mostly due to the momentum advection associated with mid-latitude baroclinic waves. This chapter provides a more detailed description of the code and numerics used in the QTCM1.

2.2 Calling Structure and Grid

2.2.1 Model Code Calling Tree

```
Driver
 |
 |
 |=> DriveInit                  (input user-control parameters)
 |
 |=> TimeManager                (starting the time manager)
 |
```

[1] p. 143

```
|=> OceanInit
|     |
|     |=> sstin or bndry1       (read initial SST)
|
|=> qtcminit
|     |
|     |=> parinit               (model parameters)
|     |     |
|     |     |=> humtable        (set up saturation humidity table)
|     |
|     |=> varinit               (initialize prognostic variables)
|     |
|     |=> bndinit               (boundary data)
|
|=> outpinit                    (open the global output files)
|     |
|   [ |=> gaCtl                 (write GrADS descriptor file) ]
|
| DO 1   (ocean-atmosphere coupling loop)
|
|=> TimeManager                 (advance the calendar)
|
|=> Ocean                       (ocean module)
|     |
|     |=> sstin                 (read in SST at the right time)
|     |
|     |=> TimeInterp            (interpolate SST)
| or
|     |
|     |=> getQflux              (read Qflux)
|     |=> getSfcHeat            (compute averaged surface fluxes)
|     |=> mxstep                (compute SST)
|
|=> QTCM                        (QTCM1 main)
|     |
|     |=> getbnd
|     |     |
|     |     |=> getSST          (fetch SST from Ocean)
|     |     |
|     |     |=> bndry1          (fetch climatological boundary data - albedo)
|     |
|     | Do 2 (land-atmosphere coupling loop)
|     |
|     |=> set_qclock            (advance the clock for the loop)
|     |
```

```
|     |=> physics1              (physics package for mode T1)
|     |     |
|     |     |=> mconvct         (moist convection)
|     |     |
|     |     |=> cloud           (compute or read cloud fraction)
|     |     |
|     |     |=> radsw           (shortwave radiation)
|     |     |     |
|     |     |     |
|     |     |     |=> solartop  (incoming solar at TOA)
|     |     |
|     |     |=> radlw           (longwave radiation)
|     |     |
|     |     |=> sflux           (surface fluxes)
|     |     |
|     |     |=> abl             (Atmospheric boundary layer model)
|     |
|     |=> sland1                (land-surface scheme)
|     |
|     |=> atmostep
|     |     |
|     |     |=> advctuv         (advection of momentum)
|     |     |     |
|     |     |=> advctTq         (advection of T, q)
|     |     |     |
|     |     |=> dffus           (diffusion)
|     |     |     |
|     |     |=> barcl           (baroclinic component V1)
|     |     |     |
|     |     |=> bartr           (barotropic component V0)
|     |           |
|     |           |=> fatdpkg   (Poisson solver)
|     |           |
|     |           |=> geopot    (diagnose geopotential)
|     |
|     |=> varmean               (summing/averaging variables)
|     |
|     |=> cplmean               (computing mean sfc. flux if coupled with ocean)
|     |
|     | ENDDO 2
|
|=> outpall
|     |
|     |=> outpinst              (output instantaneous values)
|     |
```

```
 |    |=> outpmean             (output mean values such as monthly average)
 |    |
 |    |=> out_restart          (prognostic variables for
 |                              restart run)
 |
 | ENDDO 1
 |
 |=> finish                    (wrap up)
      |
      |=> out_restart          (force out_restart, overwrite
                                old ones)
```

2.2.2 Output Calling Branches

For Sequential Unformatted Binary (GrADS) Output

Output Initialization Branch

```
driver
 |=> outpInit                                    (output initialization driver)
 |    |=> gaCtl('i')                             (write descriptor file for inst. val.)
 |    |    |=> defOutVars                        (here: Count variables)
 |    |    |    |=> defVar                       (increase variable counter)
 |    |    |=> startDate                         (compute output start date)
 |    |    |=> defOutVars                        (write the variables)
 |    |         |=> defVar                       (write one variable to descriptor)
 |    |
 |    |=> gaCtl('m')                             (write descriptor file for means)
 |    |    |=> defOutVars                        (here: Count variables)
 |    |    |    |=> defVar                       (increase variable counter)
 |    |    |=> startDate                         (compute output start date)
 |    |    |=> defOutVars                        (write the variables)
 |    |    |    |=> defVar                       (write one variable to descriptor)
 |    |    |    |    |=> SetOutVar               (set up storage for  mean field)
```

Output branch:

```
driver
 |=> outpAll                                      (write output driver)
 |    |=> outTime                                 (time for output?)
 |    |=> outpInst                                (instantaneous values)
 |    |    |=> writeI                             (write one variable)
 |    |
 |    |=> outTime                                 (time for mean output?)
```

```
 |     |=> outpMean                                 (mean value output)
 |     |     |=> writeM                             (write time means of one variable)
 |     |     |     |=> geto                         (get variable mean values)
 |     |     |     |     |=> doMean                 (compute time mean values)
```

For netCDF Output Files

Output initialization branch:

```
driver
  |=> outpInit                                      (output initialization driver)
  |     |=> defOutVars (with isinit=.true.)         (only storage allocation)
  |     |     |=> SetOutVar                         (set up storage for means)
```

Output branch:

```
driver
 |=> outpAll                                        (write output driver)
 |     |=> outTime                                  (time for inst. output?)
 |     |=> outpInst                                 (instantaneous values output)
 |     |     |=> writeI                             (write one variable)
 |     |     |     |
 |     |     |     |\ <--- first call branch
 |     |     |     | \
 |     |     |     |  |=> ncInit                    (file initialization)
 |     |     |     |  |     |=> nf_create           (create netCDF file)
 |     |     |     |  |     |=> nf_put_att_xxx      (write attributes)
 |     |     |     |  |     |=> nf_def_var          (define variables)
 |     |     |     |  |     |     |=> defOutVars (isinit=.false.) (get output variables)
 |     |     |     |  |     |     |     |=> defOutVar      (one output variable)
 |     |     |     |  |     |     |     |     |=>nf_def_var (define the variable)
 |     |     |     |  |     |     |     |     |=>nf_put_att (write variable attributes)
 |     |     |     |  |
 |     |     |     |  |=> writeStype                (write non-changing fields)
 |     |     |     |  |     |=> nf_inq_varid        (get variable netCDF ID)
 |     |     |     |  |     |=> nf_put_var_real     (write data to file)
 |     |     |     | /
 |     |     |     |/
 |     |     |     |=> nf_inq_varid                 (get variable netCDF ID)
 |     |     |     |=> nf_Close (upon error)        (close file on error)
 |     |     |     |=> nf_put_vara_real             (write variable)
 |     |
 |     |=> outTime                                  (time for mean output?)
 |     |=> outpMean                                 (mean value output)
 |     |     |=> writeM                             (write time means of one variable)
 |     |     |     |
```

```
 |     |     |     |\ <--- first call branch           (as for writeI; see above)
 |     |     |     | \
 |     |     |     |  |=> ncInit
 |     |     |     |  |=> writeStype
 |     |     |     | /
 |     |     |     |/
 |     |     |     |=> nf_inq_varid                    (get variable netCDF ID)
 |     |     |     |=> geto                            (get variable mean values)
 |     |     |     |      |=> doMean                   (compute time mean values)
 |     |     |     |
 |     |     |     |=> nf_put_vara_real                (write mean values to file)
 |     |=> out_restart                                 (write restart file)
```

Common for Binary and netCDF Output

Mean field computation branch:

```
qtcm
 |=> varmean                                           (called in main loop)
 |     |=> oacc                                        (increase counter)
 |     |     |=> sumUp                                 (sum up data)
```

2.2.3 Model Grid

The grid-scale variables in the QTCM1 are located on a staggered C-grid [28] of dimensions $NX \times NY$, where the x (zonal, longitudinal) coordinate indexes from 1 to NX, and the y (meridional, latitudinal) coordinate indexes from 1 to NY. (there are certain exceptions to this description of the dimensioning, which are noted in Section 2.3.1). A schematic of the grid is given in Figure 2.1.

The pair (i, j) denotes a grid point. For the standard version, $NX = 64$ and $NY = 42$. Figure 2.1 abbreviates variables as follows:

`T`	Temperature (e.g. `T1`)
`u`	Zonal velocity (e.g. `u0`, `u1`)
`v`	Meridional velocity (e.g. `v0`, `v1`)
`vort0`	Barotropic relative vorticity

These variables are all prognostic.

Thus, temperature is located at grid point (i, j), while zonal velocity is located at $(i + \frac{1}{2}, j)$, meridional velocity is located at $(i, j + \frac{1}{2})$, and ζ_0 is located at $(i + \frac{1}{2}, j + \frac{1}{2})$.

```
        v                   v                   v    ...     v

        |                   |                   |            |
        |                   |                   |            |

j=NY    T ------ u ------ T ------ u ------ T --- ... --- T

        |                   |                   |            |
        |                   |                   |            |

        .         .         .         .         .            .
        .         .         .         .         .    ...     .
        .         .         .         .         .            .

        |                   |                   |            |
        |                   |                   |            |

j=3     T ------ u ------ T ------ u ------ T --- ... --- T

        |                   |                   |            |
        |                   |                   |            |

        v      vort0      v      vort0      v    ...     v

        |                   |                   |            |
        |                   |                   |            |

j=2     T ------ u ------ T ------ u ------ T --- ... --- T

        |                   |                   |            |
        |                   |                   |            |

        v      vort0      v      vort0      v    ...     v

        |                   |                   |            |
        |                   |                   |            |

j=1     T ------ u ------ T ------ u ------ T --- ... --- T

        |                   |                   |            |
        |                   |                   |            |

        v                   v                   v    ...     v

        |                   |                   |            |
        |                   |                   |            |

j=0     --------------------------------------------- ... -----
       i=1                 i=2                 i=3   ...    i=NX
```

Figure 2.1: QTCM1 staggered C-grid. Note that $j = 0$ only exists for v and variables dependent on v. See text in Section 2.3.1 for details.

Recall that the subscript "0" refers to the barotropic mode, while the subscript "1" refers to the baroclinic mode.

Other variables are also defined at the above locations. These include:

T_s Located at the same position as `T`. Variable T_s is the surface temperature, and is received from the ocean, except over land where it is prognostic.

q_1 Located at the same position as `T`. Variable q_1 is the humidity, and is prognostic.

ψ_0 Located at the same position as `vort0`. Variable ψ_0 is the barotropic streamfunction, and is diagnostic from `vort0`.

All other model variables are diagnostic, and are centered at (i, j) on the grid.

Although the C-grid has half-step variable indices, array indices in FORTRAN do not. Thus, we have adopted the convention that all half-indices are truncated when referred to in the QTCM1 code. For example:

`u1(1,j)`	in the code refers to	$u_1(1\frac{1}{2}, j)$	on the C-grid
`v1(i,1)`		$v_1(i, 1\frac{1}{2})$	
`vort0(1,1)`		$\zeta_0(1\frac{1}{2}, 1\frac{1}{2})$	

In all other sections of this document, array indice naming follows the FORTRAN model code convention.

2.3 Model Code and Numerics

2.3.1 Grid Boundaries

The east/west boundaries are periodic, and the north/south boundaries are solid wall. At the north-south boundaries, meridional velocity v is set to zero. Thus, at the northern boundary, both `v0(i,NY)` and `v1(i,NY)` are set to zero. To describe the southern boundary, an extra point has been added at the southern-most extent of arrays `v0` and `v1`. Thus, the y-direction of these two arrays (and arrays which depend on them, such as `psi0`) are actually dimensioned 0:NY instead of 1:NY, as the other arrays (such as `u0` and `u1`) are. An exception is `vort0`, which one might think is dimensioned 0:NY, but actually is dimensioned 1:NY. Then, this southern-most point of both `v0` and `v1` (i.e. `v0(i,0)` and `v1(i,0)`) are set to zero. As a reminder, `v0(i,0)` and `v1(i,0)` describe $v_0(i, \frac{1}{2})$ and $v_1(i, \frac{1}{2})$, respectively. To avoid special code for for advection at the northern and southern boundaries the array holding the prognostics u_1, q_1 and T_1 are dimensioned from $j = 0$ to $j = ny + 1$.

Note that though the east/west boundaries are periodic, numerically this is implemented without the use of ghostpoints.

Between V2.0 and V2.1, there has been a shift of grid points that offsets which point occurs at the equator. This is to accommodate the pole-to-pole option in V2.1. In V2.1 and higher versions, the equator has a `v` (`v0` and `v1`) point, with `T` and `u`(`u0` and `u1`) points half a grid north and south of the equator. In V2.0 and lower, the equator has a `T` point.

2.3.2 High Latitude Filtering

Fourier filter

A Fourier filter poleward of 60° is used to filter out small scale structures. It is not turned on in versions prior to V2.3. This spatial filter of high wave numbers follows Arakawa and Lamb (1977) and uses FFT package in fatdpkg.f90.

Sponge Layers (obsolete)

Sponge layers are no longer necessary as of version 2.3 to stabilize the model. If need arises they can be turn on by defining the preprocessor macro `SPONGE`.

In the standard code v2.2, artificial ocean boundary is used and no sponge functions are turned on. To deal with the boundary instability a sponge layer can be at latitudes higher than 45° (this cutoff value is set by the local variable `y0` in subroutine `parinit`). There are five options of sponge layer functions. The first function `spngh0` acts to relax land surface temperature to climatology. The sponge function `spngh1` reduces heat and moisture flux and `spngh2` increase diffusion in the boundary layer. `spngh3` increases momentum damping, while `spngh4` reduces excessive precipitation in high latitudes. In the version 2.1 release, only `spngh0`,`spngh1` and `spngh3` are active; sponge layer terms `spngh2` and `spngh4` are commented out. The functional descriptions of the sponge layer are set in subroutine `parinit`.

Sponge layer functions are defined as:

$$\texttt{spngh0(j)} = \max\left(1, \frac{|y_j| - y_0}{10}\right)$$

$$\texttt{spngh1(j)} = \max\left(\frac{1}{2}, 1 - \frac{|y_j| - y_0}{10}\right)$$

$$\texttt{spngh2(j)} = \left[\max\left(\frac{1}{4}, 1 - \frac{|y_j| - y_0 dfs}{10}\right)\right]^{-1}$$

$$\texttt{spngh3(j)} = \left[\max\left(\frac{1}{10}, 1 - \frac{|y_j| - y_0}{10}\right)\right]^{-1}$$

$$\texttt{spngh4(j)} = \max\left(\frac{1}{10}, 1 - \frac{|y_j| - y_0}{10}\right)$$

where y_j is the latitude (in degrees) at the jth grid point. In Figure 2.1, this is the location where the `u` (and `T`) variables are located.

2.3.3 Spherical Geometry

Factors in the model equations associated with spherical geometry have been included in QTCM1 beginning with v2.0. However, in v2.0 through 2.2, these factors appeared only in the `dx` terms and were omitted for instance in $(\cos\varphi)^{-1}(\partial_\varphi(v\cos\varphi)$ term in the divergence. These have now been corrected in v2.3. The front-end subroutine for the fatd package have also been made consistent for spherical geometry.

The $R_{earth}^{-1} u \tan(\varphi)\mathbf{v}$ term of the momentum equation is neglected since it would add computation of several additional terms (such as $u_0 v_0$, etc) and is small relative to $f\mathbf{v}$ especially in the tropics.

The spherical factors are controlled in subroutine `parinit`. Array `cosu` is the cosine factor at each "whole gridpoint" latitude location, i.e. where the `u` variable is defined on the C-grid (Table 2.1). Array `cosv` is the same at the "half-grid" locations, i.e. where the `v` variable is defined on the C-grid. Arrays `dxu`, `dxv`, `dyu`, and `dyv` contain the corresponding grid distances in spherical coordinates. To remove spherical geometry effects, set both `cosu` and `cosv` to 1.

2.4 Variable Naming

2.4.1 More On Naming Conventions

Some of the basic conventions used in the code have been previously described. We describe a few more of the conventions used in naming variables below:

1. Distinguishing between barotropic vs. baroclinic variables:

   Generally, variables used to describe the barotropic mode have "0" attached to the end (e.g. `u0`, `v0`). Generally, variables used to describe the baroclinic mode have the suffix "1" attached (e.g. `u1`, `v1`).

2. Variable designation `rhs`:

   The designation "rhs" refers to "right-hand side." A variable with this designation (e.g. `rhsu1`, which is the right-hand side of the u_1 differential equation) as part of its name lumps together all the non-time-dependent terms of a particular equation. This makes it easier to increment the system a time step.

3. `im1` and `ip1`

   The variable `im1` means "variable `i` minus one." The variable `ip1` means "variable `i` plus one." These are useful for handling the periodic boundary conditions.

4. Radiation flux direction:

   Net longwave radiation flux (FLW) into an atmospheric column is defined such that if $FLW > 0$, then the whole column absorbs energy. Thus:

   $$FLW = FLW_{us} - FLW_{ds} - FLW_{ut}$$

   where "us" means upwards, at the surface; "ds" means downwards, at the surface; and "ut" means upwards, at the top-of-atmosphere.

5. `dayofmodel` variable versus function:

   The label `dayofmodel` is the name of the variable. With underscores, i.e. `day_of_model`, the term refers to the function.

Besides the prognostic and diagnostic permanent grid variables, the model also uses some temporary local variables in its computations. Below is an explanation of the names of some of these temporary variables:

u0atC Located at the same position as `u`. Variable `u0atC` represents "u_0 calculated at a 'grid center' position." Temperature (`T1`) is located at this location. Variable `u0atC` is computed as the mean of the two u_0 values one half-steps in "front" and in "back" of the "grid center." Thus, `u0atC(2,j)` $= \frac{1}{2}\left(u_0(2\frac{1}{2}, j) + u_0(1\frac{1}{2}, j)\right)$.

vatu Located at the same position as `u`. Variable `vatu` represents "v calculated at the u position," and is computed as the mean of the meridional velocity `v` at the "four corners" surrounding the `u` point location.

fu Located at the same j position as `u`. Variable `fu` represents the Coriolis parameter evaluated at the latitudinal circle `u` is located at.

fv Located at the same j position as `v`. Variable `fv` represents the Coriolis parameter evaluated at the latitudinal circle `v` is located at.

Most variables are dimensioned (NX, NY). However, because of the treatment of the boundaries, some of the model variables are dimensioned a bit differently (see discussion in Section 2.3.1).

Below is a table explaining the meaning of some variables whose function may not be obvious from the variable name and from the conventions described above. These are relatively "minor" variables (e.g. not a "main" variable like advection) that is specific only to the code, having no direct counterparts in the QTCM1 papers Neelin and Zeng [1] or Zeng et al. [2]. See Section 2.4.2 for a table relating code variables with the notation in the QTCM1 papers.

NAME	DESCRIPTION
`cosui`	Inverse of `cosu`, cosine of latitude at "whole gridpoint" locations
`cosvi`	Inverse of `cosv`, cosine of latitude at "half gridpoint" locations
`dxui`	Inverse of `dxu`, the x-direction grid spacing
`dxvi`	Inverse of `dxvi`, the x-direction grid spacing at "half gridpoints".
`dyi`	Inverse of `dy`, the y-direction grid spacing

2.4.2 Variable and Parameter Keys

The following tables provide a key relating variables and constants in the code with their counterparts in the QTCM1 papers by Neelin and Zeng [1] and Zeng et al. [2]. A list of selected definitions of some of the variables given in the QTCM1 papers follows the tables.

Note though that some of the variables given in the tables below are commented out in the released version of the code. The abbreviation "non-dim" means non-dimensional. Table I in Neelin and Zeng [1] give the values for selected parameters and coefficients. Table 3 in Zeng et al. [2] gives values for selected land parameterization coefficients.

Also, in the main QTCM1 papers [1, 2], T_1 is given in units of J kg^{-1}, while in the model, `T1` is in units of K. Thus, in this manual, everywhere you see T_1, you should assume that it includes a factor of c_p. Thus, `T1` $= c_p T_1$. Similarly, for humidity terms, a factor of L is subsumed into moisture. Thus, as a summary: `T1` $= c_p T_1$, `q1` $= L q_1$, `GMsr` $= c_p M_{Sr1}$, `GMqr` $= L M_{qr1}$, `GMsp` $= c_p M_{Sp1}$, and `GMqp` $= L M_{qp1}$.

Dynamics Variables and Parameters

CODE SYMBOL	DESCRIPTION	MODEL UNITS	PAPER SYMBOL
`a1`	Vertical profile of perturbed temperature	non-dim	a_1
`a1hat`	Vertical average of a_1	non-dim	$\hat{a}_1$
`a1p`	Coefficient a_1^+; see definition of `V1`	non-dim	a_1^+
`advT1`	Negative of advection projected onto `T1`	K/s	$-(\mathbf{v}_0 \cdot \nabla T_1 + \hat{a}_1^{-1}\langle a_1 V_1\rangle \mathbf{v}_1 \cdot \nabla T_1)$
`advq1`	Negative of advection projected onto `q1`	K/s	$-(\mathbf{v}_0 \cdot \nabla q_1 + \hat{b}_1^{-1}\langle b_1 V_1\rangle \mathbf{v}_1 \cdot \nabla q_1)$
`b1`	Vertical profile for perturbed humidity	non-dim	b_1
`b1hat`	Vertical average of `b1`	non-dim	$\hat{b}_1$
`b1s`	`b1` at surface (used in calculating evaporation)	non-dim	b_{1s}
`bb1`	Vertical profile for perturbed convective reference humidity	non-dim	B_1
`bb1hat`	Vertical average of `bb1`	non-dim	$\hat{B}_1$
`eps_0`	Damping for `v0` (barotropic component)	day^{-1}	ϵ_0
`eps_1`	Damping for `v1` (baroclinic component)	day^{-1}	ϵ_1
`eps_01`	Cross damping due to `v0` on `v1`	day^{-1}	ϵ_{01}
`eps_c`	Inverse of convective timescale τ_c	sec^{-1}	$1/\tau_c$
`GMq`	Moist stratification	K	M_{q1}
`GMqr`	Reference state moist stratification	K	M_{qr1}
`GMqp`	Perturbation to the moist stratification due to q_1, per unit q_1 change	non-dim	M_{qp1}
`GMs`	Dry stability	K	M_{S1}
`GMsr`	Reference state dry stability	K	M_{Sr1}
`GMsp`	Perturbation to the dry stability due to `T1`, per unit `T1` change	non-dim	M_{Sp1}
`Omega1`	Vertical profile of vertical velocity ω_1	non-dim	Ω_1
`Prec`	Precipitation (i.e. Q_c)[a]	W/m^2	Q_c
`psi0`	Barotropic streamfunction	m^2/s	ψ_0
`q1`	Humidity	K	q_1

[a]The model code expresses precipitation in energy units. However, in a few equations in the QTCM1 papers, precipitation is given as P, which is often given in water mass units. The precipitation variable given here is equivalent to the former description of precipitation.

CODE SYMBOL	DESCRIPTION	MODEL UNITS	PAPER SYMBOL
`qrefs`	Reference humidity at surface	K	q_r at surface
`T1`	Air temperature	K	T_1
`Trefs`	Reference air temperature at surface	K	T_r at surface
`Tsref`	Reference surface temperature	K	T_{sr}
`Ts`	Surface temperature: for ocean is SST, for land is ground temperature	K	T_s
`u0`	Zonal wind, barotropic mode	m/s	u_0
`u1`	Zonal wind, baroclinic mode	m/s	u_1
`v0`	Meridional wind, barotropic mode	m/s	v_0
`v1`	Meridional wind, baroclinic mode	m/s	v_1
`V1`	Vertical profile of baroclinic velocity, where $V_1(p) = a_1^+ - \widehat{a_1^+}$	non-dim	V_1
`V12hat`	Vertical average of V_1^2	non-dim	$(\widehat{V_1^2})$
`V1s`	`V1` at the surface	non-dim	$V_{1\mathrm{s}}$
`vort0`	Barotropic vorticity	s^{-1}	ζ_0

Surface Flux Variables and Parameters

CODE SYMBOL	DESCRIPTION	MODEL UNITS	PAPER SYMBOL
`Cp`	Specific heat of air at constant pressure	J kg^{-1} K^{-1}	c_p
`Cpg`	Factor to convert heat flux into column heating rate	J K^{-1} m^{-2}	$c_p p_T/g$
`Evap`	Evaporation	W/m^2	E
`Hlatent`	Latent heat of evaporation	J kg^{-1}	L
`qs`	Saturation humidity at T_s	K	$q_{sat}(T_s)$
`taux`	Surface stress along x	N/m^2	x-component of τ_s
`tauy`	Surface stress along y	N/m^2	y-component of τ_s
`ps`	Surface Geopotential	HPa	ϕ_s
`dphisdx`	Zonal derivative of surface geopotential	$mathrmHPa/m$	$\frac{\partial \phi_s}{\partial x}$
`dphisdy`	Zonal derivative of surface geopotential	$mathrmHPa/m$	$\frac{\partial \phi_s}{\partial y}$
`ub`	Zonal wind at the top of the ABL	m/s	
`vb`	Meridional wind at the top of the ABL	m/s	
`us`	Zonal surface wind	m/s	
`vs`	Meridional surface wind	m/s	
`weml`	Entrainment velocity at top of ABL	m/s	
`ziml`	Fixed height of the ABL	m/s	
`Wsmin`	Minimum wind speed for surface flux computation	m/s	

Land-Surface Variables and Parameters

CODE SYMBOL	DESCRIPTION	MODEL UNITS	PAPER SYMBOL
`albdveg`	Surface albedo	fraction	A_s
`beta`	Surface beta factor	fraction	β
`rsmin`	Minimum bulk surface resistance (including stomatal/root resistance), occurring at no water stress ($\beta = 1$)	m^{-1} s	$r_{\mathrm{s}_{\mathrm{min}}}$
`Runf`	Runoff	$\mathrm{W/m^2}$	R
`STYPE`	Surface type (ocean = 0, land > 0)		
`WD`	Soil moisture content in terms of equivalent water depth[a]	$\mathrm{kg/m^2}$	W
`xla`	Leaf area index	non-dim	LAI
`WD0`	Soil moisture field capacity	mm	W_0
`Z0`	Roughness length	m	Z_0

[a] 1 $\mathrm{kg/m^2}$ equals 1 mm equivalent depth for the case of liquid water, if we assume a density of 1000 $\mathrm{kg/m^3}$.

Radiation Variables and Parameters

CODE SYMBOL	DESCRIPTION	MODEL UNITS	PAPER SYMBOL
`cl3`	Cloud amount	fraction	α_n, for n=3, where n specifies cloud type. The cloud type parameterizes on deep convection. [a]
`FLWds`	Downward longwave flux at surface	W/m^2	$R_s^{\downarrow}$
`FLWus`	Upward longwave flux at surface	W/m^2	$R_s^{\uparrow}$
`FSWds`	Downward shortwave flux at surface	W/m^2	$S_s^{\downarrow}$
`FSWus`	Upward shortwave flux at surface	W/m^2	$S_s^{\uparrow}$
`FSWut`	Upward shortwave flux at TOA[b]	W/m^2	$S_t^{\uparrow}$
`FTs`	Sensible heat flux	W/m^2	H
`OLR`	`FLWut` in code, upward longwave flux at TOA	W/m^2	$R_t^{\uparrow}$
`S0`	Incoming solar radiation at TOA[c]	W/m^2	$S_t^{\downarrow}$

[a] In clrad0, this is high and middle clouds. In clrad1 and clrad1-d, these are deep and CsCc clouds.

[b] TOA = "top of atmosphere"

[c] In the QTCM1 papers [1, 2], S_0 refers to the solar constant.

Selected QTCM1 Variable and Parameter Definitions

The following definitions of the "paper variables" are given in Neelin and Zeng [1] and Zeng et al. [2]. We present them here as an aid in interpreting the above variable keys.

Parameter definitions:

$$\hat{a}_1 = p_T^{-1} \int_{p_{rt}}^{p_{rs}} a_1 \, dp, \quad \hat{b}_1 = p_T^{-1} \int_{p_{rt}}^{p_{rs}} b_1 \, dp$$

$$M_1 = M_{S1} - M_{q1}$$

$$\begin{aligned} M_{S1} &= MSr1 + M_{Sp1}T_1 \\ &= p_T^{-1} \int_{p_{rt}}^{p_{rs}} \Omega_1(-\partial_p s)\, dp \end{aligned}$$

$$\begin{aligned} M_{q1} &= Mqr1 + M_{qp1}q_1 \\ &= p_T^{-1} \int_{p_{rt}}^{p_{rs}} \Omega_1 \partial_p q\, dp \end{aligned}$$

$$\begin{aligned} \Omega_1(p) &= -\int_p^{p_{rs}} V(\grave{p})\, d\grave{p} \\ &= -\int_p^{p_s} \left(a_1^+(\grave{p}) - \widehat{a_1^+} \right) d\grave{p} \end{aligned}$$

where M_1 is the gross moist stability, and $s = T + \phi$ is the dry static energy. Recall that a c_p term has been absorbed into T and that L is subsumed in the q_1 terms in the Neelin and Zeng [1] and Zeng et al. [2] equations.

Temperature, moisture, and momentum variable definitions:

$$T = T_r(p) + a_1(p)T_1(x, y, t)$$

$$q = q_r(p) + b_1(p)q_1(x, y, t)$$

$$q_1 = \frac{\int_{p_s}^{p_t} q \cdot a_1(p)\, dp}{\int_{p_s}^{p_t} a_1(p)\, dp}$$

$$T_1 = \frac{\int_{p_s}^{p_t} T \cdot a_1(p)\, dp}{\int_{p_s}^{p_t} a_1(p)\, dp}$$

$$\mathbf{v} = \mathbf{v_0}(x, y, t) + V_1(p)\mathbf{v_1}(x, y, t)$$

$$\mathbf{v}_0 = (\mathrm{u}_0, \mathrm{v}_0), \quad \mathbf{v}_1 = (\mathrm{u}_1, \mathrm{v}_1)$$

Recall that the vertical averaging operator is defined as follows in Neelin and Zeng [1]:

$$\widehat{X} = \langle X \rangle = p_T^{-1} \int_{p_{rt}}^{p_{rs}} X\, dp$$

Below are alternative (but equivalent) expressions for a few of the code variables given in the tables in Section 2.4.2:

$$\texttt{advT1} = -\frac{\int_{p_s}^{p_t} \mathbf{v}(p) \cdot \nabla T(p)\, a_1(p)\, dp}{\int_{p_s}^{p_t} a_1(p)\, dp}$$

$$\texttt{advq1} = -\frac{\int_{p_s}^{p_t} \mathbf{v}(p) \cdot \nabla q(p)\, a_1(p)\, dp}{\int_{p_s}^{p_t} a_1(p)\, dp}$$

Some final notes: p_T is the depth of the troposphere (in pressure units), and g is the gravitational acceleration at the Earth's surface.

2.5 Coupling Considerations

The routines found in cplmean.f are designed for the purpose of coupling the QTCM1 atmosphere with an ocean model. Fluxes in time are averaged over the coupling interval in cplmean.f.

N.B. the call of cplmean is turned off by default. A defined preprocessor macro `CPLMEAN` turns it on. Cplmean is design to provide the surface heat and momentum fluxes. Taking cplmean as a guideline it should be easy to compute other average surface fluxes as needed.

QTCM1 reads the sea surface temperature by a call of getSST in the file getbnd.f. This routine as well as the driver may have to be adjusted to meet the coupling needs. Indeed, the driver can be cannibalized to fit calls into another model driver if convenient.

2.6 Dealing With Model Blow-up

Although the standard version of model has been throughly tested, users may experience occasional model numerical instability and "blow-up" when grid resolution or domain size is changed, or model physical parameters are modified. For example, the model may sometimes produce strong high latitude disturbances, especially over land; or the model may sometimes create a grid-scale disturbance over tropical land that produces excessive land cooling and precipitation. If the model warms up too much over all (for instance in a coupled simulation) it tends to be prone to blow up unless the changes in dry static stability are properly represented. Another problematic region could be summer monsoon area, where strong gradients of precipitation can occur along the coast of China at the land-sea contrast. Possible reasons for the blow-up include that the model physical parameterizations have certain range of validity, and that the model numerics is chosen to achieve maximum computational efficiency under current settings. When different parameters are used, the validity of model assumptions may be violated or the CFL criterion is exceeded at certain locations. Very often there is a combination of pushing the model into a range of questionable validity in some part of the domain (sometimes within the sponge layer), followed by numerical instability. We considered adding fail-safe checks at various places in the code, but the computational cost and added code complexity seemed to make this sub-optimal, since most users are familiar with computational instability.

If you experience a blow-up, it is best to diagnose the conditions under which it seems to be occurring and what is different from the standard run. Modifying the physical parameters in qtcmpar.in, changing the use of sponge boundary functions, changing the horizontal viscosity coefficients, or capping the largest coefficient of momentum advection can help work around a particular instability. Reducing the time step may seem obvious, but may not help if the model is encountering one particular region where the parameterizations are being pushed beyond their range of validity. Users are welcome to report difficulty to resolve blow-ups to the QTCM development team. We will be happy to share our experience with you. However, we cannot guarantee to debug each individual case.

A useful feature implemented in v2.3 is that the model would generate a file of instantaneous values of standard variables when the model blows up. The mean for the last month right before blowing up is also written in a monthly output file.

Chapter 3

Model Physics

The main references for the model formulation are papers by Neelin and Zeng [1] and Zeng, et al. [2]. A summary of the project leading up to the initial QTCM1 formulation may be found in Neelin [7], while the basis for the analytical approximations for dynamics are found in Neelin and Yu [8] and Yu and Neelin [13]. Documentation of the radiation and cloud packages may be found in Chou and Neelin [5] and Chou [3]. A partial summary of the model formulation is provided as Appendix A.

This chapter summarizes some of the parameterizations used to describe atmospheric physics, as a well as implementation and discussion of the justifications used in calculating key parameters used in QTCM1.

3.1 Calculation of the Parameters in qtcmpar.f90

A summary of the variables in this file is given in Section 1.4.1. The parameters in qtcmpar.in are generated by a Fortran 90 program, par.f90. To recalculate qtcmpar.f90 (and qtcmpar.gs) uncomment the build lines in the makefile and execute `make qtcmpar.f90` It produces the file, qtcmpar.f90. The input reference profile for par.f is Tqo3.ref which includes SST (K), air temperature (K), water vapor mixing ratio (kg/kg), and ozone mixing ratio (kg/kg) in 1 hPa interval from 1000 hPa to 1 hPa.

For calculating `a1`, 900 hPa cloud base and 150 hPa cloud top are used. Below the cloud base, the atmosphere is dry adiabatic; above the cloud base, the atmosphere is moist adiabatic. Relative humidity profile is a linear fit to the observed reference relative humidity, which is 85% near surface and 60% at 150 hpa. Tqo3.ref is used to calculate reference values of the gross moist stability and the gross moisture stratification. However, for some sensitive quantities, the values from the single profile in qtcmpar.f90 are used as a guideline balanced with other considerations. In particular, reference values of the gross dry stability and gross moisture stratification, GMsr and GMqr, are assigned in qtcmpar.in based after sensitivity testing. Relevant factors include: (i) the choice of dry stability affects numerical stability. Sometimes a larger value of this parameter is useful to avoid sporadic blow-up after decades of run. (ii) the moisture tends to adjust nonlinearly in the model so the climatological value in the warm pool region may be somewhat off the values in the reference profile. (iii) Estimates of the gross moist stability M in @CROSSREF[11] differed in absolute value among different data sets, but the relative value of M/Ms was more consistent and is more dynamically important. There is thus some freedom to choose GMsr and GMqr within known constraints to attempt to obtain better numerical stability and reasonable output values of M and GMq. In version 2.3 GMsr=3.5 and GMqr=3.0.

Model top in the calculation of all basis functions was set at 200mb in v2.2 and earlier. This was partly motivated by the values of V_1 that result above 200 mb from the contribution of T_1/p in the vertical integration of the hydrostatic equation and the impact of these in $\langle V_1^2 \rangle$ and $\langle V_1^3 \rangle$ coefficients. In v2.3, the velocity basis function V_1 is held constant above 280 mb and the model top can thus be extended to 150mb, more typical of the tropical troposphere. Since temperature at the uppermost levels is not strongly held to quasi-equilibrium, it is reasonable to cap V_1 values there to prevent impact in the higher order terms. This impacts momentum transport terms, including effects of the vertical advection of momentum.

3.2 Land-Surface Parameterization Scheme

The land-surface parameterization used in QTCM1 is fully described by Zeng et al. [2]. Here we briefly describe the methodology. The scheme used is only moderately more complicated than the bucket model, and is termed Simple-Land, or simply SLand. It does not attempt to resolve accurately the diurnal solar and environmental control on photosynthesis. Thus the soil moisture and seasonal variation of radiation are the main controlling factors. The most essential features for climate simulation are the low heat capacity of the land surface, specification of land albedo for surface energy budget, and soil moisture and its consequences for surface water budget. In SLand, a single soil layer is assumed but with different depth for the energy and the water balance. Subgrid-scale variability of rainfall can significantly influence surface runoff and interception loss, and therefore evaporation. Various analytical formulations have been proposed (e.g., Entekhabi and Eagleson [22]). SLand follows

	Ocean	Forest	Grass	Desert
Surface albedo A_s	0.07	0.12	0.19	0.3
Min. resistance $r_{s_{\min}}$ ($\mathrm{m}^{-1}\mathrm{s}$)		150	200	200
Field capacity W_0 (mm)		500	400	300
Roughness length Z_0 (m)	0.0024	2.0	0.1	0.05
Leaf area index		6	3	1

Table 3.1: Parameter values used in the land-surface model (consolidated from BATS and SIB2). The listed albedo values are not used in the standard version of QTCM1; see the text in Section 3.2 for more details.

essentially the same statistical approach but with choices more in line with the level of complexity of QTCM1.

Although the scheme allows many land-surface types as long as the relevant parameter values are provided, we opt for a simple classification in the standard version with three surface types: forest, grass, and desert, while more surface types can be used in applications emphasizing land-surface processes. Surface type is specified in the array `STYPE`, which is of type REAL∗8. The most important surface properties include surface albedo and field capacity which play critical roles in the energy and the water balance, respectively. The important surface-type dependent parameter values, including albedo, minimum surface resistance, field capacity, surface roughness and leaf area index, are listed in Table 3.1 (which is identical to Table 3 in Zeng et al. [2]). In the standard version of QTCM1, we use prescribed surface albedo derived from satellite observations (Darnell et al. [21]), while one can easily switch to using the surface albedos linked to surface type listed in Table 3.1. Sensitivity studies show discernible differences at regional scales between the two methods, but these have little impact on the global patterns. Snow hydrology is not simulated in this version, since the atmospheric model is aimed at the tropics.

In v2.3, the interception loss $Evapi$ is capped at 50% of precipitation. This is because when occasionally precipitation reaches extreme high values, a chain of land surface feedbacks is triggered. $Evapi$ could exceed $Prec$ due to abnormal $Evapi_o$ (based on net non-evaporative surface flux) and is associated with very cold T_s, eventually causing the model to blow up. Under normal conditions, $Evapi$ is approximately linear with $Prec$. Over stype forest, $Evapi$ is close to $0.17Prec$ and is smaller for other surface types. The specified cap is seldom reached and helps model numerical stability.

3.3 Cloud-Radiation Parameterization Coefficients

This section describes the algorithms used in computing the coefficients in clrad1.f and clrad1d.f.

3.3.1 Cloud Fraction

Three cloud types defined by the ISCCP C product are used: deep cloud combined with CsCc (cloud type 1), cirrus (cloud type 2), and stratus (cloud type 3). CsCc linearly depends on deep cloud, so we combine them to be one cloud type. Cloud type 1 linearly depends on large-scale precipitation with the coefficient:

$$\texttt{cl1P} = 7.76 \times 10^{-4}\ (\mathrm{W\ m^{-2}})^{-1}$$

For more details, see Section 2.2 in Zeng et al. [2].

In version 2.3, a cap is implemented to ensure cloud 1 fraction α_1 does not exceed 1 (note in theoretical studies this nonlinearity can be neglected but not in the numerical version especially with higher variability as in v2.3). Additional conditions are applied to other cloud types to account for overlap.

Let α_k^{tot} be the total cloud fraction of cloud type k before taking account of overlap and α_k be the cloud fraction that enters the radiation code after taking overlap into account (code variables `cldtot(k,i,j)` and `cld(k,i,j)`). The sum of α_k is required to be less than or equal to 1 with the clear sky fraction (α_0) equal to the remainder. Since overlapping clouds are not typically observed by satellite, the α_k fractions are more directly comparable to ISCCP data as well as being the radiatively active cloud fraction in the model. When cloud type 1 overlaps any other cloud type, its radiative properties dominate so

$$\alpha_1 = \alpha_1^{tot}$$

with

$$\alpha_1^{tot} = \texttt{min(cl1P Qc,1)}$$

In version 2.3, cirrus cloud fraction is parameterized on Cloud Type 1 as

$$\alpha_2 = \alpha_2^{tot}(1 - \alpha_1)$$

with

$$\alpha_2^{tot} = \texttt{min}(\texttt{cl2fac} * \alpha_1, 1)$$

The proportionality factor `cl2fac`=1.5, based on ISCCP C2 data (Chou 1997, Fig. 5.15 and taking into account the combination of deep and CsCc in cloud type 1) although we note there is considerable scatter about this regression in the data. This cloud fraction of Cloud 2 is further modified by a cloud overlap condition to ensure total cloud fraction less than 1. The physical assumption is that where overlap occurs the deep cloud effect on radiation dominates so α_1 is used in the radiation. The $(1-\alpha_1)$ factor is equivalent to a random overlap assumption. The cloud 2 fraction that results from this parameterization in v2.3 results has slightly larger maximum cloud fraction in convergence zones than ISCCP and smaller fraction in the subtropics. Spatial averages within the tropics are similar to observed for 10S-10N but somewhat smaller than observed for 30S-30N.

For Cloud 3 and Cloud 4, we use

$$\alpha_3 = \alpha_3^{tot}(1 - \alpha_1 - \alpha_2)$$

$$\alpha_4 = \alpha_4^{tot}(1 - \alpha_1 - \alpha_2)$$

such that when Cloud 3 or Cloud 4 is shielded by Cloud 1 or Cloud 2, the Cloud 1 or Cloud 2 properties are used. For stratus, a spatial constant or observed seasonal climatology from the ISCCP C2 data can be used for α_3^{tot}. Note that if observed stratus is used it is modified by the overlap factor of $(1 - \alpha_1 - \alpha_2)$ if precipitation occurs in a grid box with stratus. If using a constant reference value α_{kr}, $\alpha_3^{tot} = \alpha_{3r}/(1 - \alpha_{1r} - \alpha_{2r})$ helps keep spatial mean similar to α_{3r}. For Cloud 4, we choose not to read in an observed climatology since in the ISCCP data these cloud types do not have an obvious spatial and seasonal dependence. In previous versions, cloud type 4 was strictly constant and its cloud fraction was absorbed into the radiation code parameters to reduce computation. Here cloud type 4 is variable due to overlap. In addition to overlap with cloud types 1 and 2, cloud type 4 is reduced if cloud type 3 is large:

$$\alpha_4^{tot} = \mathtt{min}(\alpha_{4r}/(1 - \alpha_{1r} - \alpha_{2r})\,,\ (1 - \alpha_3^{tot})).$$

The condition $\sum_{i=1}^{4} \alpha_i \leq 1$ with these parameterizations factorizes to $\alpha_1 \leq 1$, $\alpha_2^{tot} \leq 1$, and $\alpha_3^{tot} + \alpha_4^{tot} \leq 1$, which all have been enforced by the `min` function conditions in the parameterizations.

With this parameterization, the modeled type 1 cloud fraction is about 0.5-0.6 for intense storms. The ISCCP climatology for deep cloud plus CsCc has maximum around 0.5 as well.

3.3.2 Longwave Radiation

Coefficients of Table 1 in Zeng et al. [2] are calculated by the Chou and Neelin weakly nonlinear scheme [5] with a reference profile, Tqo3.ref. One can use a command `run coef.sh`, and the coefficients will be in lw-coef.dat. The reference values of radiative fluxes are also calculated. The Harshvardhan longwave radiation (lwdrive.f) is also included for calculating the reference radiative fluxes and the Green's functions.

3.3.3 Shortwave Radiation

Coefficients of Table 2 in Zeng et al. [2] are calculated by a simplified shortwave radiation scheme modified from the Fu-Liou [23] shortwave radiation scheme, and the reference profile is Tqo3.ref. One can use a command `run coef.sh`, and the coefficients will be in sw-coef.dat. The Fu-Liou scheme (sw.f) is also included. With radiative flux data files, fsw.dat and fswd.dat, there is no need to re-run the Fu-Liou shortwave radiation scheme which often takes some time to run unless cloud optical properties have been changed.

3.4 Slab Mixed-layer Ocean Model

The slab mixed-layer ocean model assumes a fixed mixed-layer depth (default value is 50 meters). The SST is determined by the energy balance between radiative flux at surface, latent and sensible heat fluxes and Q-flux, which crudely simulates the ocean transport. Q-flux is typically taken from a control QTCM1 run when SST is specified as observations or climatology. Or estimates of observed net flux may be used. In either case, the implied ocean heat transport is calculated by the balance between net surface fluxes and rate of change of SST. It can be annually averaged or monthly averaged. The air-sea coupling interval is 1 day, i.e., SST is updated every 24 hours. To activate the mixed-layer ocean model, use the pre-processor macro "MXL_OCEAN".

A variant of mixed-layer ocean model is to combine it with prescribed observed SST in specific regions, defined by SST mask file. This option is implemented by defining the preprocessor macro "BLEND_SST". When the SST mask value is 1, the read-in observed SST is used. When the SST mask is 0, SST computed from the mixed-layer ocean is used.

3.5 Convective Closure Assumptions

Different from previous versions, V2.2, by default, uses a "nonlinear q_c" scheme for convective closure assumption. It is in principle the same as the original Betts-Miller convective closure scheme but retains the nonlinear Clausius-Clapeyron dependence of q^c (the convective moisture profile) on temperature in horizontal variations as well as vertical. In the original scheme, linearization is used about a reference profile at each level, which holds well through the tropics, but is beyond its range of validity in mid-latitudes in winter. The nonlinear scheme significantly reduces the excessive mid-latitude precipitation shown in the original model and prevents occurrence of negative humidity in mid-latitudes. The original linear scheme is commented out in `qtcm.F`.

Another convective closure assumption implemented but commented out in `qtcm.F` is called the 'unshifted-T^c scheme' or 'U-T^c scheme', which assumes that the convective temperature profile, T^c, arises following a moist adiabat from the boundary layer without the shift used in the original Betts-Miller scheme to match the moisture sinks. This condition is indeed applied to the moisture sink term. It avoids explicitly specifying convective moisture profile. Simulations with this scheme produce similar results to the original linear scheme.

3.6 Horizontal Diffusion Schemes

Two types of diffusions are used to control nonlinear instability and aliasing. The second-order diffusion is used for temperature and moisture equations with diffusion coefficients `viscT` and `viscQ`, respectively. A more scale-selective fourth-oder diffusion is used in the momentum equations with a tunable diffusion parameter `visc4U`. The second-order diffusion scheme takes the form

$$F_A = K_H \nabla^2 A \tag{3.1}$$

and center differencing is used for $\nabla^2 A$. The fourth-order diffusion is of the form

$$F_A = {K_H}'(\partial_x^4 + \partial_y^4)A \tag{3.2}$$

and a nine-point differencing is used following the scheme used in the PSU/NCAR MM5. With an isotropic x and y coefficients, and absorbing a factor of (grid space)$^{-2}$, this is

$$\begin{aligned} F_A(i,j) \;=\;& \mathrm{visc4/dx^2}\Big[A(i+2,j) + A(i-2,j) \\ & -4\Big(A(i+1,j) + A(i-1,j)\Big) + 6A(i,j)\Big] \\ +\;& \mathrm{visc4/dy^2}\Big[A(i,j+2) + A(i,j-2) \\ & -4\Big(A(i,j+1) + A(i,j-1)\Big) + 6A(i,j))\Big]. \end{aligned} \tag{3.3}$$

The nine points used (labeled by their weights) for this scheme are as follows (for the diffusion on the center point).

```
0  0  1  0  0

0  0 -4  0  0

1 -4  6 -4  1

0  0 -4  0  0

0  0  1  0  0
```

In version 2.3, spherical geometry is considered for the differential equations. In horizontal diffusion schemes, the weights for the points in the difference stencil satisfy $\sum_{j=2}^{NY-2}(cos(j)\sum_{k=1}^{5}(weight4(j,k)A(j+k-3))) = 0$. for any field A, and when $cos(j) = 1$, the weights are the same as in Cartesian Coordinate shown above.

The use of the fourth-order diffusion scheme was originally motivated by the desire to have diffusion terms be negligible in the zonal and vertical average momentum budget (relative to the balance of surface stress and nonlinear momentum transport). Downgradient diffusion is a poorer approximation for momentum than for a conserved tracers, hence the decision to try and keep the impacts of horizontal diffusion at large scales small via fourth-order representation in the momentum equations, while retaining second order in temperature and moisture equations. The values of the diffusion parameters were tuned largely on the trade-off of lowering diffusion but keeping numerical stability in long runs. The lower diffusion also tends to increase the transient eddy variance which tends to improve surface stress climatology.

3.7 Atmospheric Boundary Layer

QTCM version 2.3 includes a simple atmospheric boundary layer. It assumes a steady state, vertically homogeneous mixed layer of fixed height z_i. This reduces the momentum equation in the boundary layer to a simple balance between the Coriolis force due to the boundary layer wind $\mathbf{u}_s = (u_s, v_s)$, the boundary layer's average gradient of the geopotential ϕ_s and the turbulent stress $\tau = (\tau^x, \tau^y)$ at top and bottom (see Stevens et al. [10]):

$$f\mathbf{k} \times \mathbf{u}_s = -\nabla\phi + \frac{\tau(z_i) - \tau(0)}{z_i}. \tag{3.4}$$

The stress $\tau(z_i)$ at the top of the mixed layer is assumed to be proportional to the velocity jump at the interface $\tau(z_i) = w_e(\mathbf{u}_b - \mathbf{u}_s)$ with an entrainment velocity w_e as its proportionality constant. For the surface stress the wind speed $U_s = \|\mathbf{u}_s\|$ in the standard bulk formula is augmented by a minimum surface wind speed $U_{\min}$ which crudely parameterizes the additional stress due to surface wind variance not captured in QTCM (gustiness etc.): $\tau(0) = C_D U_s \mathbf{u}$, $U_s = \sqrt{U_{\min}^2 + u_s^2 + v_s^2}$. The result is a nonlinear vector equation for the boundary layer wind. In components:

$$0 = fv_s - \frac{\partial\phi}{\partial x} + \frac{w_e}{z_i}(u_b - u_s) - \frac{C_D}{z_i}U_s u, \tag{3.5}$$

$$0 = -fu_s - \frac{\partial\phi}{\partial y} + \frac{w_e}{z_i}(v_b - v_s) - \frac{C_D}{z_i}U_s v. \tag{3.6}$$

These equations are solved for u_s and v_s in subroutine `abl` using a Newton-Ralphon procedure. The routine uses the surface wind of the prior time step as the initial guess. Initially, the routine uses the (u_b, v_b) as the initiall guess[1]. The input of `abl` are the surface geopotential gradient $\nabla\phi_s$ and the velocity at the top of the ABL $\mathbf{u}_b$. Besides u_s and v_s `abl` returns U_s which is then used in the bulk formulas for the latent and sensible heat fluxes.

3.8 Cloud Top Effect on Gross Moist Stability

Over deep convective regions, high cloud tops associated with strong convection would increase gross dry stability, and thus gross moist stability, because of increased thickness of cloudy layers. In v2.3, this cloud top effect is included by a crude parameterization based on strong correlation of boundary layer moisture and convective activity in the atmosphere. Comparing to the constant GMs used in previous versions, GMs takes the following form in v2.3,

$$GMs = GMsr + GMsq * max(q1(i,j), q1m), \tag{3.7}$$

where $GMsq$ is the coefficient of gross dry stability dependence on moisture. We use the value of $GMqp$ for $GMsq$ such that the gross moist stability M would be constant over moist regions. Observational evidence shows that these two parameters are similar. We choose them equal for simplicity. The threshold value of surface

[1] The surface wind is currently not saved in the restart file which may make the model not exactly restartable, i. e., the trajectories of a restarted run may diverge from the original trajectory.

moisture is decided by looking at model output and picking the typical value of moisture in warm SST regions. The default is set to be 19 g kg^{-1}.

The inclusion of cloud-top effect on GMs helps to prevent gross moist stability from decreasing lower than minimum value typical of the reference profile region. It also helps numerical stability. Without this, it is typical that one has to raise $GMsr$ for numerical reasons.

3.9 Notes on model simulation for various versions

Archived sample figures of QTCM simulations for various versions are available from `http://www.atmos.ucla.edu/~csi/QTCM/`, along with comparisons to the observed climatology for your reference. For a number of climate features, you may notice that the differences between simulations by different versions are modest changes in intensity, slight shifts in position of particular features such as convection zones, storm tracks, transition to westerlies, etc. The notes here simply point out some aspects that we have noted as particular to each version. This section was added with v2.3, so notes on earlier versions are sparse.

V2.3 Released August 2002

- Many of the changes between v2.2 and v2.3 were motivated by ocean coupling. Considerable attention was paid to the climatological surface winds and stress which tended to be too strong in v2.2. These are now weaker in the tropics, in better agreement with observations. We note that the zonal average wind stress is controlled by nonlinear momentum transports, so this quantity is quite a challenging test. Winds at 850mb are much less sensitive: a change in the smaller surface winds is less noticable at this level or higher. For the simulation at midlatitudes, the northern hemisphere surface wind remains too strong and the southern hemisphere too weak for their respective winter seasons.
- As a trade-off, the ENSO wind stress anomalies are weaker than in v2.2, and it has been difficult to strengthen these without also increasing the climatological stress.
- A number of physics changes have contributed to the surface wind configuration. The inclusion of vertical advection of momentum contributes substantially to surface wind changes. The change of the boundaries to 80S-80N impacts the midlatitude transients which in turn improves both

tropical and extratropical winds. The change to fourth order diffusion reduces the effects of viscosity at large scales in the momentum equations. The atmospheric boundary layer for winds influences the wind turning and especially the equatorial wind strength.

- When the ABL is turned off, the overall climatology is quite similar. Extratropical precipitation tends to be slightly stronger when the ABL is on. The zonal mean zonal wind stress is very similar but the zonally asymmetric zonal stress has differences, notably along the equator where the ABL reduces the magnitude of the stress.
- The gross moist stability is smaller (more in line with the estimated range from observations or reanalysis data sets). This results in somewhat more intense convection zones and increased precipitation anomalies in interannual variabilty, including ENSO.
- There is more tropical intraseasonal variability in this version than in the last few versions (even without inclusion of the stochastic precipitation parameterization explored in Lin and Neelin 2000 in v2.2). In time series, this variability looks promising, but a spectral wavenumber decomposition shows that the release version differs from observations in some spectral characteristics and has too much westward propagating variance at low wavenumbers along the equator. This behavior is sensitive to parameters including the gross dry stability reference value and the minimum wind speed in evaporation. Anyone interested in examining the intraseasonal variability might consider exploring for an improved regime.

V2.2 Released November 2000

- This version has very low tropical intraseasonal variability. This can actually be an asset for certain purposes. For instance, Lin and Neelin (2000) used it to test what intraseasonal variability could be maintained by a stochastic convective parameterization, with the control having little variability maintained by other mechanisms.

Appendix A

Summary of Model Formulation

Contents modified from: Neelin, J. D., and N. Zeng, 1997: The simplest quasi-equilibrium tropical circulation model—formulation. *Proceedings of the Seventh Conference on Climate Variations (Climate 7)*, 2-7 February, 1997, Long Beach, California, Amer. Met. Soc., pp. 4–7.

If some notational differences remain between this summary and the Neelin and Zeng *J. Atmos. Sci.* article [1] the latter should be considered the authoritative version.

A.1 Introduction

Simple models of the tropical atmosphere (Gill 1980, Webster 1981, Zebiak 1986, Lindzen and Nigam 1987, Neelin and Held 1987, Wang and Li 1993) have proven useful for a number of problems, especially those where it is sufficient to obtain low-level wind from a given latent heating or sea surface temperature (SST). These models make different assumptions about convection, and there is an unresolved debate about the relative importance of competing mechanisms by which the SST pattern can potentially influence the circulation. However, the models behave similarly (Neelin 1989) and have a number of tunable parameters that are sufficiently ill-constrained that the models can be fit to the low-level wind field (e.g., Allen and Davey 1993). To resolve such ambiguities, and to approach problems involving radiative transfer, surface heat exchange, cloud effects and the stability of the tropical climatology, it becomes more important to have models intermediate between these and general circulation models (GCMs). Seager and Zebiak (1995) take a numerical approach to this problem, using a linear primitive equation model with the Betts-Miller (1986) scheme.

We outline here a very different approach in which we parlay theoretical results into a numerical scheme. We have analytical results for vertical structure that hold under certain approximations in convective regions (Neelin and Yu 1994; Yu and Neelin 1994). These result from using the constraints on temperature that occur in quasi-equilibrium convective closures. By using these as the leading basis functions in a Galerkin representation of vertical structure, we get a model that gives back the theoretical results in simple cases, but is valid more generally. Because the main benefits depend on the use of quasi-equilibrium constraints, we refer to these models as "quasi-equilibrium tropical circulation models" (QTCMs).

We present here the simplest model version, along with a summary of the general approach to deriving these models.

A.2 QTCM Approach

All QTCM versions are derived from the nonlinear primitive equations

$$(\partial_t + \mathcal{D}_T)T + \omega\partial_p s = Q_c + g\partial_p R^{\uparrow} - g\partial_p R^{\downarrow} - g\partial_p S + g\partial_p F_T \tag{A.1}$$

$$(\partial_t + \mathcal{D}_q)q + \omega\partial_p q = Q_q + g\partial_p F_q \tag{A.2}$$

$$(\partial_t + \mathcal{D}_V)\mathbf{v} + f\mathbf{k}\times\mathbf{v} + g\partial_p\tau = -\nabla\int_p^{p_{rs}} \kappa T\, d\ln p - \nabla\phi_s \tag{A.3}$$

$$\omega = \omega_s + \int_p^{p_s} \nabla\cdot\mathbf{v}dp \tag{A.4}$$

$$\omega_s \approx -\rho_a g\mathbf{v}_s\cdot\nabla z_s \tag{A.5}$$

where temperature T and specific humidity q are in energy units, $R^{\downarrow}$, $R^{\uparrow}$ are the long-wave radiative fluxes depending on T, q and cloudiness over the column, and surface temperature T_s. Shortwave fluxes S are positive downward. The operators $\mathcal{D}_T$, $\mathcal{D}_q$ and $\mathcal{D}_v$ include horizontal diffusion and horizontal advection terms. The vertical fluxes of sensible heat and moisture by nonconvective, diffusive transport, F_T and F_q, and stress τ, vanish at the top and have drag laws at the surface for F_{Ts}, evaporation E, and τ_s. The hydrostatic equation has been used for baroclinic pressure gradients in the momentum equation (A.3), where ϕ_0 is the geopotential at the lowest pressure level p_0. z_s is surface elevation, p_s is surface pressure, ρ_a is atmospheric near-surface density, and $\rho_a d\phi_s/dt$ has been neglected.

An important subsidiary relation that holds for all convective parameterizations is the "convective heating" and "moistening" terms Q_c and Q_q cannot change the vertically integrated moist static energy budget, i.e. $\widehat{Q}_c + \widehat{Q}_q = 0$ or

$$\partial_t(\widehat{T} + \widehat{q}) + \widehat{\mathcal{D}_T T} + \widehat{\mathcal{D}_q q} + \widehat{w\partial_p h} = F^{NET} \tag{A.6}$$

where $h = s + q,\ s = T + \phi,\ F^{NET} = S_t - S_s - R_t^{\uparrow} - R_s^{\downarrow} + R^{\uparrow sfc} + E + F_{Ts}$, absorbing $(g/\Delta p_t)$ into all fluxes for brevity, and $\widehat{(\ )} = \Delta p_t^{-1} \int_{p_s}^{p_t} (\ ) dp$ denotes vertical averaging over the troposphere, from the surface, p_s, to the tropopause, p_t.

A.2.1 *Basis Functions*

The Galerkin vertical representation of temperature in terms of basis functions $a_k(p)$ is:

$$T = T_r(p) + \sum_{k=1}^{K} a_k(p)\, T_k(x, y, t) \tag{A.7}$$

where $T_k(x, y)$ are coefficients of the expansion in p chosen to minimize square error under an inner product $\langle\ \rangle$ that is just a weighted pressure average. We split off a reference profile $\overline{T}$ that does not vary in space because this can be specified or treated in a separate calculation—only gradients of T affect the circulation. Note that $\overline{T}$ is *not* assumed to be a solution of the system, and that the spatial average of the model output differs from $\overline{T}$.

The first trick that aids both numerics and interpretation is that basis functions for the wind fields are constructed from approximate analytic solutions. Consider the leading temperature basis function, $a_1(p)$. Temporarily neglecting $\partial_p \tau$ and advection by baroclinic wind, the solution to the momentum equation (A.3) for circulations driven by this baroclinic pressure gradient, satisfying simple lower and upper boundary conditions, has the form

$$\begin{aligned} \mathbf{v}(x, y, p, t) &= V(p)\mathbf{v}_T(x, y, t) \\ V(p) &= (a_1^+(p) - \widehat{a_1^+}) \\ a_1^+(p) &= \int_p^{p_{rs}} a_1(\grave{p}) d\ln \grave{p} \end{aligned} \tag{A.8}$$

Using such analytically constructed basis functions to derive nonlinear Galerkin equations can be highly accurate at low truncation, and provides a physical association of $\mathbf{v}$ and T components.

The approach (A.7)-(A.8) could be used generally but larger gains occur when the temperature variations that are typically encountered are restricted by convection. In our approach, we *tailor the temperature basis functions* to typical quasi-equilibrium profiles. One of the main effects of convection on the large scales is to constrain the vertical temperature profile through the depth of the convecting region—highly unstable profiles with large convective available potential energy tend to be modified by convection (e.g., Xu and Emanuel 1989; Randall and Wang 1992). For many convection schemes one can explicitly or implicitly define, for a given set of large scale variables, a quasi-equilibrium temperature profile towards which convection brings the large scale temperature (e.g., Manabe et al 1965; Arakawa and Schubert 1974; Betts 1986). Thus the variations in quasi-equilibrium temperature profile can be efficiently expressed in the form (A.7) but with only N coefficients $T_k^{(QE)}(x,y)$, where $N \leq K$ is small. The coefficients $T_k^{(QE)}$ depend on the large scale T and q variables. If a_1 is typical of temperature variations near deep convective regions, associated velocity variations will be largely carried in V_1. Similarly for shallow convection, and so on.

A.3 The Simplest QTCM

The simplest QTCM, referred to as QTCM1.0, occurs when we carry only a single temperature basis function typical of deep convection effects. Two leading basis functions are carried for the velocity field: V_0 for the barotropic mode and V_1 given by (A.8) for the baroclinic mode associated with T_1 variations. Projecting on these gives equations that need be solved only in (x,y) for $\mathbf{v}_0(x,y)$ and $\mathbf{v}_1(x,y)$.

A.3.1 *Momentum Equations*

$$\partial_t \mathbf{v}_1 + \mathcal{D}_{V1}(\mathbf{v}_0, \mathbf{v}_1) + f\mathbf{k} \times \mathbf{v}_1 = -\kappa \nabla T_1 - \epsilon_1 \mathbf{v}_1 - \epsilon_{01} \mathbf{v}_0 \tag{A.9}$$

$$\partial_t \zeta_0 + \mathrm{curl}_z \left(\mathcal{D}_{V0}(\mathbf{v}_0, \mathbf{v}_1) \right) + f \nabla \cdot \mathbf{v}_0 + \beta v_0 = -\mathrm{curl}_z(\epsilon_0 \mathbf{v}_0) - \mathrm{curl}_z(\epsilon_{10} \mathbf{v}_1) \tag{A.10}$$

with

$$\nabla \cdot \mathbf{v}_0 = -\omega_s / \Delta p_t$$

where the operators $\mathcal{D}_{m1}$ etc include nonlinear advection terms by retained velocity components. Most importantly, the effects of vertical momentum transfer have been included, with

$$\epsilon_0 = \frac{g}{\Delta p_t} \rho c_D \mathrm{V}_s \ ,$$

$$\epsilon_1 = \frac{g}{\Delta p_t} \rho c_D \mathrm{V}_s V_{1s}^2 / \langle V_1^2 \rangle + \langle \nu a_1^2 p^{-2} \rangle / \langle V_1^2 \rangle$$

incorporating both spin down by surface stress and internal viscous momentum transfer within the baroclinic mode. Both are nonlinear functions of $\mathbf{v}_0$ and $\mathbf{v}_1$ through the surface wind speed V_s.

A.3.2 *Moist Static Energy Equation and the Gross Moist Stability*

The moist static energy equation is

$$\widehat{a}_1(\partial_t + \mathcal{D}_{T1})T_1 + \widehat{b}_1(\partial_t + \mathcal{D}_{q1})q_1 + M_1 \nabla \cdot \mathbf{v}_1 = (g/p_T)F^{\mathrm{net}} \tag{A.11}$$

where the *gross moist stability*

$$M_1 = \int_{p_t}^{p_s} \Omega_1 \partial_p h dp \tag{A.12}$$

acts as an effective stability for the system in moist regions. Note that $\Omega_1(p) = \int_p^{p_s} V_1 dp$, the vertical velocity profile corresponding to the velocity profile V_1, is given by (A.8).

We have specified nothing about the convective closure on moisture so far, only aspects of the closure on temperature. In convective regions, all that is needed from the convective closure for (A.12) is the vertical mean moisture and surface moisture (related to T_1 and the large scale circulation). Thus the essence of the model does not place any demands on the particulars of the convective closure on moisture, nor on the numerical treatment of moisture.

A.3.3 *Temperature and Moisture Equations*

The thermodynamic equation and moisture equations, for temperature and moisture projections T_1 and q_1 associated with basis functions $a_1(p)$ and $b_1(p)$ are:

$$\widehat{a_1}(\partial_t + \mathcal{D}_{T1})T_1 + M_{S1}\nabla \cdot \mathbf{v}_1 = \langle Q_c \rangle + (g/p_T)(-R_t^{\uparrow} - R_s^{\downarrow} + R_s^{\uparrow} + S_t - S_s + H) \tag{A.13}$$

$$\widehat{b_1}(\partial_t + \mathcal{D}_{q1})q_1 - M_{q1}\nabla \cdot \mathbf{v}_1 = \langle Q_q \rangle + (g/p_T)E \tag{A.14}$$

For QTCM1, we use a version of the Betts-Miller (1986) moist convective adjustment scheme, with convection restoring temperature toward convective profile T^c. The moisture sink and convective heating terms are given by

$$-\langle Q_q \rangle = \langle Q_c \rangle = \epsilon_c \left[(\widehat{T_r^c} + \widehat{a_1} T_1^c) - (\widehat{T_r} + \widehat{a_1} T_1 \Big) \right] \tag{A.15}$$

where $\epsilon_c = \tau_c^{-1}\left(\widehat{a_1}\widehat{b_1}/(\widehat{a_1}+\widehat{b_1})\right)\mathcal{H}_c(\mathrm{C})$, with τ_c the time scale of convection in strongly convecting regions and $\mathcal{H}_c(\mathrm{C})$ a smoothed Heaviside function depending on C_1, given in square brackets, a quantity related to CAPE projected on retained structures. The convective profile of temperature is given as T_r^c a reference profile independent of space, plus departures T_1^c following a moist adiabat from the boundary layer with vertical profile of $a_1(p)$. An additional closure assumption is required, which links T_1^c to T_1 and q_1 after some manipulation. Examples of this closure are discussed in [1] where the dry stability M_{S1} and the gross moisture stratification M_{q1} are the contributions of s and q to M_1 in (A.12). The sum of these equations yields the moist static energy equations (A.11) with the q terms simplified to $(\widehat{b}_1\partial_t + D_{q1})q_1$.

A.4 Land-Surface Model

We use a single land-surface layer

$$C_s\partial_t T_s = F_s^{NET} \tag{A.16}$$

where C_s is a land heat capacity; since this is small, on time scales much longer than a day, the condition is essentially zero net surface flux.

Evaporation over land is modified as

$$E = \rho\left[q_{sat}(T_s) - q_a\right](r_a + r_s)^{-1} \tag{A.17}$$

where $r_a = (C_D\mathrm{V})^{-1}$ is the aerodynamic resistance; r_s is similar to the combined stomatal and root resistance (Dickinson 1984), and increases as the soil wetness w drops. Evaporation thus becomes less dependent on wind speed and roughness for low soil moisture, akin to biophysical models.

For interactive soil wetness, we use a single layer formulation representing the root zone

$$\partial_t w = P - E - R \tag{A.18}$$

with P the precipitation and R the runoff.

A.5 Discussion

Equations (A.9)-(A.10) and (A.13)-(A.14) are the main prognostic equations for QTCM1.0. Although they need only be solved in x, y and t, the resulting flow solutions are three dimensional. The conceptual point of view implied by these equations differs from the traditional description of diabatic heating driving circulation, which feeds back onto the heating via moisture convergence. In regions in convective quasi-equilibrium, the system (A.9), (A.10) and (A.11) is closed. The moist static energy budget (A.11) governs the thermodynamics of convective regions, with (A.13) mainly serving to establish how far the large scale dynamics can push the CAPE out of quasi-equilibrium and how big the heating is. If ϵ_c is large compared to other time scales in the system, (A.13) is essentially diagnostic for the heating. As one passes from convective to non-convective regions (A.13) takes over; in between gradually larger departures from quasi-equilibrium occur. Convective heating and precipitation are by-products of a moist dynamics that evolves subject to convective constraints on baroclinic pressure gradients in convective regions. The gross moist stability M_1 in (A.11) thus determines the phase speed of the Madden-Julian oscillation, the magnitude of the circulation response to SST anomalies, and the radius of deformation for motions in deep convective regions. Yu et al (1997) discuss the climatology of the gross moist stability, estimated from radiosonde data and ECMWF analyses.

Over oceans, the net heating into the atmospheric column, F^{NET} in (A.11), has terms in E, F_{Ts}, and longwave fluxes that act as forcing by SST and terms that depend on temperature and moisture that act as a damping to perturbations. Over land, when (A.16) is near equilibrium, the surface flux vanishes, so the net flux input on the rhs of (A.11) becomes

$$F^{NET} = F_t^{NET} = S_t - R_t^{\uparrow} \; .$$

Since the outgoing longwave radiation $R_t^{\uparrow}$ depends on the atmospheric state, the solar input is clearly identified as the forcing of the system, but with important modifications by cloud albedo feedbacks in S_t. Diagnostics based on the moist static energy budget thus have an attractive simplicity, especially over land, in convective regions.

The progression from the simplest QTCM described above to more complex versions consists mainly of adding successive basis functions corresponding to additional physics, particularly shallow convection and boundary layer variations not linked to moist convection.

A.6 References

Allen, M.R., and Davey, M.K. 1993: Empirical parameterization of tropical ocean-atmosphere coupling: the inverse Gill problem. *J. Clim.,* **6**, 509-530.

Arakawa, A., and W. H. Schubert, 1974: Interaction of a cumulus cloud ensemble with the large-scale environment, Part I. *J. Atmos. Sci.*, **31**, 674-701.

Betts, A. K., 1986: A new convective adjustment scheme. Part I: Observational and theoretical basis. *Quart. J. R. Met. Soc.*, **112**, 677-691.

Betts, A. K., and M. J. Miller, 1986: A new convective adjustment scheme. Part II: Single column tests using GATE wave, BOMEX, ATEX and arctic air-mass data sets. *Quart. J. R. Met. Soc.*, **112**, 693-709.

Dickinson, R. E., 1984: Modeling evapo-transpiration for three-dimensional global climate models. In *Climate processes and climate sensitivity*, J. E. Hanson and T. Takahashi (Eds.), Geophysical monograph 29, 5, Amer. Geophys. Union, 58-72.

Gill, A. E., 1980: Some simple solutions for heat induced tropical circulation. *Quart. J. Roy. Meteor. Soc.*, **106**, 447-462.

Lindzen, R. S., and S. Nigam, 1987: On the role of sea surface temperature gradients in forcing low-level winds and convergence in the tropics. *J. Atmos. Sci.,* , **45**, 2440-2458.

Manabe, S., J. Smagorinsky and R. F. Strickler, 1965: Simulated climatology of a general circulation model with a hydrological cycle. *Mon. Wea. Rev.*, **93**, 769-798.

Neelin, J. D., and I. M. Held, 1987: Modeling tropical convergence based on the moist static energy budget. *Mon. Wea. Rev.*, **115**, 3-12.

Neelin, J. D., 1989: On the interpretation of the Gill model. *J. Atmos. Sci.*, **46**, 2466-2468.

Neelin, J. D., and J.-Y. Yu, 1994: Modes of tropical variability under convective adjustment and the Madden-Julian oscillation. Part I: Analytical results. *J. Atmos. Sci.*, **51**, 1876-1894.

Randall, D.A., and J. Wang, 1992: The moist available energy of a conditionally unstable atmosphere. *J. Atmos. Sci.,* **49**, 240-255.

Seager, R., and S. E. Zebiak, 1995: Simulation of tropical climate with a linear primitive equation model. *J. Clim.*, **8**, 2497-2520.

Wang, B., and T. Li, 1993: A simple tropical atmospheric model of relevance to short-term climate variation. *J. Atmos. Sci.,* , **50**, 260-284.

Webster, P. J., 1981: Mechanisms determining the atmospheric response to sea surface temperature anomalies. *J. Atmos. Sci.*, **38**, 554-571.

Xu, K., and K. A. Emanuel, 1989: Is the tropical atmosphere conditionally unstable? *Mon. Wea. Rev.*, **117**, 1471-1479.

Yu, J.-Y., and J. D. Neelin, 1994: Modes of tropical variability under convective adjustment and the Madden-Julian oscillation. Part II: Numerical results. *J. Atmos. Sci.*, **51**, 1895–1914.

Zebiak, S. E., 1986: Atmospheric convergence feedback in a simple model for El Niño. *Mon. Wea. Rev.*, **114**, 1263-1271.

Appendix B

Notes

This section is used for notes on: things that haven't made it into the main version of the manual yet; comments on a version noted after release; bug fixes; informal but useful information; etc.

- *On surface solar in v2.1 (noted 8/99):* In v2.1, using clrad1, while the net surface solar appears generally close to such data sets as Darnell et al. (1992), Li et al. (1993) and Bishop and Rossow (1991), it does have a systematic bias when run with only 3 cloud types (even when these are specified from ISCCP). Net surface solar tends to be about 20-30 W/m^2 higher than the Darnell et al. dataset over tropical oceans. This was noted when the QTCM1 was coupled with an ocean model (NCAR upper ocean model) and tended to have a warm SST bias. Including a term that mimics the average effects of the omitted "exotic" cloud types (e.g. AsAc), which individually do not contribute greatly to reflected solar, but tend to add up, and including aerosols in the Fu-Liou scheme estimation brings the surface solar into line with the other surface solar estimates. This modification is not included in the standard v2.1 release. Effects have not been fully analyzed as of 9/1/99.

Bibliography

[The Main Initial Papers on the QTCM1]

[1] Neelin, J. D., and N. Zeng, 2000: A quasi-equilibrium tropical circulation model—formulation. *J. Atmos. Sci.*, 1741-1766.

[2] Zeng, N., J. D. Neelin, and C. Chou, 2000: A quasi-equilibrium tropical circulation model—implementation and simulation. *J. Atmos. Sci.*, 1767-1796.

[Other Papers With Relevance to the QTCM1]

[3] Chou, C., 1997: Simplified Radiation and Convection Treatments for Large-Scale Tropical Atmospheric Modeling. Ph.D. Dissertation, University of California, Los Angeles. 215 pp.

[4] Chou, C., and J. D. Neelin, 1996: Linearization of a long-wave radiation scheme for intermediate tropical atmospheric models. *J. Geophys. Res.*, **101**, 15129–15145.

[5] Chou, C., and J. D. Neelin, 1997: Cloudiness and radiation parameterizations for intermediate tropical atmospheric models. *Proceedings of the Seventh Conference on Climate Variations (Climate 7)*, 2–7 February, 1997, Long Beach, California, Amer. Met. Soc., pp. 253–256.

[6] Lin, J. W.-B., J. D. Neelin, and N. Zeng, 1999: "Maintenance of tropical intraseasonal variability in an intermediate-level atmospheric model," *J. Atmos. Sci.*, (accepted).

[7] Neelin, J. D., 1997: Implications of convective quasi-equilibrium for the large-scale flow. In *The Physics and Parameterization of Moist Atmospheric Convection.* Ed. R. K. Smith, Kluwer Academic Publishers, Dortrecht, The Netherlands.

[8] Neelin, J. D., and J.-Y. Yu, 1994: Modes of tropical variability under convective adjustment and the Madden-Julian oscillation. Part I: Analytical results. *J. Atmos. Sci.*, **51**, 1876–1894.

[9] Neelin, J. D., and N. Zeng, 1997: The simplest quasi-equilibrium tropical circulation model—formulation. *Proceedings of the Seventh Conference on Climate Variations (Climate 7)*, 2-7 February, 1997, Long Beach, California, Amer. Met. Soc., pp. 4–7.

[10] Stevens, B., A.S. Ackerman, B.A. Albrecht, A.R. Brown, A. Chlond, J. Cuxart, P.G. Duynkerke, D.C. Macvean, R.A.J. Neggers, Neggers, E.S. Sanchez, A.P. Siebesma, D.E. Stevens, 2001: Simulations of trade wind cumuli under a strong inversion. *J. Atmos. Sci.*, **58**, 1870-1891.

[11] Yu, J.-Y., C. Chou and J. D. Neelin, 1998: Estimating the gross moist stability of the tropical atmosphere. *J. Atmos. Sci.*, **55**, 1354–1372.

[12] Yu, J.-Y., and J. D. Neelin, 1994: Modes of tropical variability under convective adjustment and the Madden-Julian oscillation. Part II: Numerical results. *J. Atmos. Sci.*, **51**, 1895–1914.

[13] Yu, J.-Y., and J. D. Neelin, 1997: Analytic approximations for moist convectively adjusted regions. *J. Atmos. Sci.*, **54**, 1054–1063.

[14] Zeng, N., and J. D. Neelin, 1997: The simplest quasi-equilibrium tropical circulation model—initial results. *Proceedings of the Seventh Conference on Climate Variations (Climate 7)*, 2–7 February, 1997, Long Beach, California, Amer. Met. Soc., pp. 8–10.

[15] Zeng, N., J. D. Neelin, C. Chou, J. W.-B. Lin, H. Su, 1998: Climate and variability in the first quasi-equilibrium tropical circulation model. General Circulation Model Development: Past, Present, and Future, D. A. Randall, ed., Academic Press. 457–488.

[**Other References**]

[16] Arakawa, A., and W. H. Schubert, 1974: Interaction of a cumulus cloud ensemble with the large-scale environment, Part I. *J. Atmos. Sci.*, **31**, 674–701.

[17] Betts, A. K., 1986: A new convective adjustment scheme. Part I: Observational and theoretical basis. *Quart. J. R. Met. Soc.*, **112**, 677–691.

[18] Betts, A. K., and M. J. Miller, 1986: A new convective adjustment scheme. Part II: Single column tests using GATE wave, BOMEX, ATEX and arctic air-mass data sets. *Quart. J. R. Met. Soc.*, **112**, 693–709.

[19] Betts, A. K., and M. J. Miller, 1993: The Betts-Miller scheme. Chapter 9 in *The Representation of Cumulus Convection in Numerical Models of the Atmosphere.* (Eds. K. A. Emanuel and D. J. Raymond) *Amer. Meteor. Soc., Meteor. Mon.* **24** No. 46, 107–121.

[20] Bishop, J. K. B., and W. B. Rossow, 1991: Spatial and temporal variability of global surface solar irradiance. *J. Geophys. Res.* **96**, 16839-16858.

[21] Darnell, W. L., W. F. Staylor, S. K. Gupta, N. A. Ritchey, and A. C. Wilber, 1992: Seasonal variation of surface radiation budget derived from international satellite cloud climatology project C1 data. *J. Geophys. Res.* **97**, 15741–15760.

[22] Entekhabi, D., and P. S. Eagleson, 1989: Land surface hydrology parameterization for atmospheric general circulation models including subgrid scale spatial variability. *J. Clim.*, **2**, 816–831.

[23] Fu, Q., and K. N. Liou, 1993: Parameterization of the radiative properties of cirrus clouds. *J. Atmos. Sci.*, **50**, 2008–2025.

[24] Haltiner, G. J., and R. T. Williams: *Numerical Prediction and Dynamic Meteorology*, New York, John Wiley & Sons, 477 pp.

[25] Li, Z., H, G. Leighton, K. Masuda, and T. Takashima, 1993: Estimation of SW Flux Absorbed at the Surface From Total Reflected Flux. *J. Clim.*, **6**, 317-330

[26] Manabe, S., J. Smagorinsky and R. F. Strickler, 1965: Simulated climatology of a general circulation model with a hydrological cycle. *Mon. Wea. Rev.,* **93,** 769–798.

[27] Manabe, S., and R. F. Strickler, 1964: Thermal equilibrium of the atmosphere with a convective adjustment. *J. Atmos. Sci.,* **21,** 361–385.

[28] Mesinger F., and A. Arakawa, 1976: *Numerical Methods Used in Atmospheric Models. Vol. 1*, GARP Publications Series No. 17, World Meteorological Organization, 64 pp.

[29] Reynolds, R. W., and T. M. Smith, 1994: Improved global sea surface temperature analyses using optimum interpolation. *J. Climate*, **7**, 929–948.