Terminal Area Simulation System User's Guide —Version 10.0

George F. Switzer
Analytical Services and Materials, Inc., Hampton, Virginia

Fred H. Proctor
Langley Research Center, Hampton, Virginia

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 443-757-5803

- Phone the NASA STI Information Desk at 443-757-5802

- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

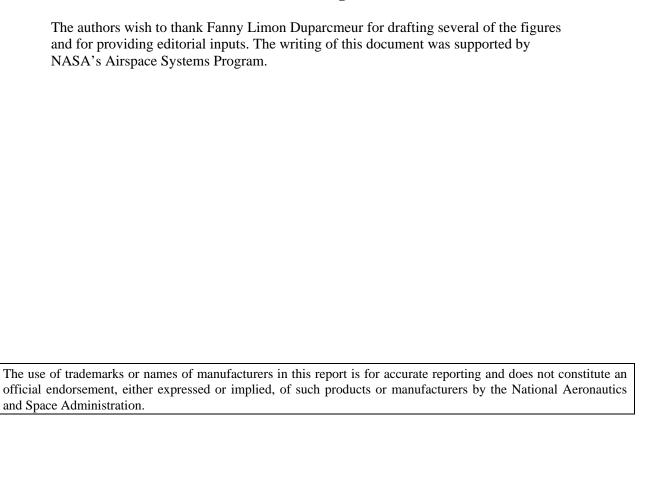# Terminal Area Simulation System User's Guide —Version 10.0

George F. Switzer
Analytical Services and Materials, Inc., Hampton, Virginia

Fred H. Proctor
Langley Research Center, Hampton, Virginia

Acknowledgments

The authors wish to thank Fanny Limon Duparcmeur for drafting several of the figures and for providing editorial inputs. The writing of this document was supported by NASA's Airspace Systems Program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Terminal Area Simulation System (TASS) is a three-dimensional, time-dependent, large eddy simulation model that has been developed for studies of wake vortex and weather hazards to aviation, along with other atmospheric turbulence, and cloud-scale weather phenomenology. This document describes the source code for TASS version 10.0 and provides users with documentation needed to run the model. The source code is programmed in Fortran language and is formulated to take advantage of vector and efficient multi-processor scaling for execution on massively-parallel supercomputer clusters. The code contains different initialization modules allowing the study of aircraft wake vortex interaction with the atmosphere and ground, atmospheric turbulence, atmospheric boundary layers, precipitating convective clouds, hail storms, gust fronts, microburst windshear, supercell and mesoscale convective systems, tornadic storms, and ring vortices. The model is able to operate in either two- or three-dimensions with equations numerically formulated on a Cartesian grid. The primary output from the TASS is time-dependent domain fields generated by the prognostic equations and diagnosed variables. This document will enable a user to understand the general logic of TASS, and will show how to configure and initialize the model domain. Also described are the formats of the input and output files, as well as the parameters that control the input and output.

THIS PAGE INTENTIONALLY LEFT BLANK

1.0 INTRODUCTION

The primary purpose of this document is three-fold: 1) to give a general description of the source code structure, 2) to explain simulation capabilities, and 3) describe internal and external parameters that control and select the type of simulation. The appendix contains a list of papers resulting from TASS application to research topics.

The structure of the document is first to briefly describe the TASS model by giving an overview addressing the governing equations, reference frame and numeric model followed by detailed descriptions of the subroutine structure of the source code. Next, the document explains the necessary input files necessary to run the model including an explanation of the internal parameters that affect operational outcomes. Next is a description of the data output from the simulation including various roles played by output files depending upon the TASS operational mode. The document concludes with an appendix containing a list of publications resulting from TASS application to research topics. The information in this document will enable a user who is familiar with both Fortran programming language and operating systems for High End Computing, to apply TASS to a variety of atmospheric and aviation related problems.

2.0 TASS MODEL CODE

The Terminal Area Simulation System (TASS) is a three-dimensional, large-eddy numerical model that has a meteorological reference frame with a compressible, non-Boussinesq equation set. TASS includes initialization modules that allow the simulation of various phenomenologies, including maritime and continental convective clouds, hailstorms, tornadic supercells, mesoscale convective complexes, atmospheric turbulence, microbursts, aircraft wake vortices, gust fronts, planetary boundary layers, and other related atmospheric events. A partial list of studies that have utilized TASS results is contained in the appendix.

TASS contains library calls for the Message Passing Interface (MPI), enabling TASS to operate efficiently on massively-parallel computer architectures. Invoking TASS requires a specification of the number of computer CPUs available. TASS arranges these CPUs, or nodes, as specified by the user and splits/decomposes the global domain into smaller sub-domains. Then each node computes a share of the simulation for one of these sub-domains. The domain decomposition is transparent to the user as the output files all represent the global domain. Only in the event the user needs to modify the code will knowledge of MPI become critical.

The MPI sub-domain structure is created to avoid degrading the highly-efficient vector logic that was present in earlier versions that existed prior to MPI implementation. This translates to a highly efficient code that can take advantage of the many core architectures present today, including those that employ vector acceleration logic. Therefore, TASS is poised to continue to take advantage of modern computer advancements in both inter-process communication and vector operations.

2.1   TASS Reference Frame

The TASS reference frame is a three-dimensional, Cartesian coordinate system aligned along the compass directions as shown in figure 2-1. The shaded portion of the figure represents the earth's surface. The horizontal location of the origin can move to track a storm or important phenomena within the domain, and the vertical axis is always the height above the earth's surface. The X and Y axes align with the East and North directions, respectively. If desired, the coordinate system in TASS can be rotated such that the X and Y coordinates lie in other compass directions as well. When configured as a two-dimensional model, the active axes are along the X and Z coordinates.

Unless noted, TASS utilizes dimensional variables and constants that assume MKS units following the System International (SI) standard (length= meters (*m*), time = seconds (*s*), mass = kilograms (*kg*), pressure = Pascals (*Pa*), temperature = Kelvin (*K*), speed = meters per second (*m/s*), *etc.*). Non-SI units sometimes used within the input or output from TASS are millibars (*mb*) for pressure, degrees Celsius for temperature (^{o}C), and inches per hour for rainfall rate.

Figure 2-1. Three-dimensional reference frame for TASS.

2.2 Prognostic TASS Equation Set and Variables

The numerical model is comprised of up to twelve prognostic equations. These include three equations for momentum, one each for pressure deviation and potential temperature, and six prognostic equations for continuity of the following water substances: water vapor, cloud droplet water, cloud ice crystals, rain, snow, and hail. The final prognostic equation is for a massless tracer. This last equation also can be utilized for dust, smoke, or insects. A listing of the prognostic variables is summarized in Table 2-1.

Phase changes for water substance, including the formation of cloud and precipitation substances are accounted for by the TASS cloud microphysics. Variables for water substance are represented by mixing ratios; i.e., *kg* of water per *kg* of dry air.

Table 2-1. Prognostic TASS Variables

Variable	Description	Units
U	Eastward component of wind velocity	m/s
V	Northward component of wind velocity	m/s
W	Vertical component of wind velocity	m/s
P	pressure deviation	Pa
θ	Potential temperature	K
DUST	Massless tracer	kg/kg
Q_{SN}	Snow water	kg/kg
Q_R	Rain water	kg/kg
Q_H	Hail water	kg/kg
Q_{CD}	Liquid cloud water	kg/kg
Q_{IC}	Ice crystal water	kg/kg
Qv	Water vapor	kg/kg

2.3 Numerical Model

The following list describes the salient features of the TASS model. Further details concerning the TASS code and the planetary boundary layer model can be found in Proctor (1987, 1996, and 2009) and Schowalter et al. (1996), respectively.

$\Rightarrow$ **Equations** - Primitive non-Boussinesq, non-hydrostatic, compressible, time-dependent.

$\Rightarrow$ **Geometry** - Either two-dimensional (slab) or three-dimensional domain. Ground is assumed flat.

$\Rightarrow$ **Microphysics** - Liquid, ice, and vapor phases: can simulate growth process for cloud droplets, cloud ice crystals, rain, and hail/graupel. Inverse-exponential size distributions are assumed for precipitating hydrometeors.

$\Rightarrow$ **Prognostic Variables** - Cartesian wind components, pressure deviation from ambient, potential temperature, water substances (water vapor, cloud droplet water, cloud ice crystals, rain, snow and hail), and massless tracers.

$\Rightarrow$ **Coordinate and Grid System** - Constant grid spacing in the horizontal directions with either stretched or constant vertical grid spacing. Arakawa C-grid staggered mesh (Haltiner 1980) is used.

$\Rightarrow$ **Time Integration** - The time integration is broken into small and large time steps. Only the acoustically active terms in the equations of momentum and pressure are computed

over the small time step (Proctor 1987, pg 78). Momentum and pressure use second-order Adams-Bashforth or first-order modified Adams-Bashforth for both large and small time steps (Switzer 1996). The temporal and spatial discretization for temperature, water substance, and DUST equations utilize Upstream-Biased Quadratic Interpolation that is third-order in space and time (Leonard 1979). Periodic hole filling is applied to remove negative water. As an option to avoid the latter, the Monotone Upstream-Centered Scheme for Conservation Laws (MUSCL) after van Leer (1979), may be used to avoid the generation of negative water (Ahmad and Proctor 2011).

⇒ **Spatial Discretization** f**or Momentum and Pressure** - Explicit, centered fourth-order quadratic conservative differences. An option is available for centered second-order derivatives. Diffusion terms are always approximated with centered second-order derivatives.

⇒ **Initial State** - Horizontally-homogeneous base state, initialized with a single vertical profile of temperature, humidity, and wind velocity. Initial perturbations may then be superimposed upon the base state. Specific initialization modules are available for simulation of convective storms, resolved-scale turbulence, microbursts, fronts, ring vortices, isolated thermal, planetary boundary layers, and aircraft wake vortices.

⇒ **Lateral Boundary Conditions** - Options for open, mirror, or periodic boundaries. Open conditions utilize mass conservative, non-reflective radiation boundary scheme. The option for a mirror condition can be only applied on the south boundary (i.e., $Q(b+\Delta y)=Q(b-\Delta y)$ where the boundary is at $y=b$ and Q is any variable).

⇒ **Bottom Boundary Conditions** - Rigid, flat, closed, no-slip bottom boundary with ground stress as a function of surface roughness height. A closed, free slip condition also is allowed as an option. Other options are available for ground heating rate, heat and moisture fluxes, and a surface energy budget. The surface energy budget option is based on a two-layer soil model, which calculates soil moisture, soil temperature, and the resulting heat and moisture fluxes to the atmosphere. As an alternative to a closed boundary condition, periodic boundaries can be applied at the bottom and top boundary.

⇒ **Top Boundary Conditions** - Option for either closed or periodic conditions. If closed, free-slip conditions are assumed, and a filter and sponge are applied to the top four rows to diminish gravity wave reflection.

⇒ **Restart Option** - Ability to resume a computation from a previous run at intervals.

⇒ **Domain Translation** - Option to translate the domain with either a constant or variable velocity. Tracking algorithm allows domain to move with the phenomena being modeled. Reduces need for domains with large horizontal extents.

⇒ **Subgrid Scale Turbulence** - Large-eddy simulation model with first-order, subgrid-scale turbulence closure. Scales of turbulence larger than grid volume are resolved in the domain. Choice of Smagorinsky model or Vreman's (2004) model. Both are modified for Richardson number dependence. Rotational damping of subgrid turbulence allowed as an option.

⇒ **Direct Numerical Simulations (DNS)** - Subgrid scale turbulence can be turned off, and a constant value for viscosity can be specified.

⇒ **Atmospheric Scale** - Most adapted for simulation of meso-γ and micro scale weather phenomenon.

⇒ **Terrain** - Flat ground with surface fluxes dependent upon local derivatives and surface roughness parameter.

2.4 TASS Subroutine Descriptions and Locations

The TASS code is subdivided into 17 modules, with each module containing subroutines that perform related functions. The source code for TASS has adapted to the changing Fortran language standards over time. Therefore, the current source has elements of Fortran syntax ranging from "Fortran 77" to "Fortran 2008." The use of the module terminology is not to be confused with the module programming units introduced in the more recent Fortran standards. This section describes the source code by starting with the overview of the individual modules followed by a detailed description of the subroutines and functions contained within each module.

2.4.1 Source Code Structure

This section will list the name and individual function of each subroutine or function subroutine contained within each module. Table 2-2 contains general descriptions of each of the 17 modules that comprise TASS. Module names are written in lowercase italics, and subroutines names are underlined in all capital letters. Some of the subroutine names appear to have spaces, but these are underscore characters hidden by the underline.

Table 2-2. TASS Module Descriptions

Module Name	Module Function
driver.f	Serves to control the execution of the overall code.
advectl.f	Computes advection and diffusion (convection) of prognostic scalar equations. Updates all prognostic scalar variables for large time step.
advects.f	Computes advection and diffusion for prognostic momentum equations. Called at each large time step.
allcomm.f	Contains all of the Message Passing Interface (MPI) subroutines.
bounds.f	Sets most boundary conditions.
dataproc.f	Selects and processes data for output.
decomp.f	Decomposes the physical domain into sub-domains for each node/processor of the run and computes the necessary vectors for inter-process communications.
diagvar.f	Computes diagnostic terms such as eddy viscosity, buoyancy, and divergence. Also outputs monitoring data from the TASS run and checks the values of prognostic variables to abort a rapidly diverging simulation. Called for each large time step.
initial.f	Initializes the domain, and sets up any necessary data areas needed for the run.
initper.f	Initializes thermal and velocity perturbation and wake vortex fields.
io.f	Contains subroutines for input and output of all non-restart related data.
march.f	Updates time integration terms necessary for stability, monitors the movement of the domain for tracking, and updates the velocity and pressure, based on acoustically active terms for each small time step.
micro.f	Computes cloud microphysics, including condensation and evaporation. Also computes hydrometeor terminal velocities. Called for each large time step.
restart.f	Performs the input and output of the restart information.
sensor.f	Similar to *dataproc.f* but designed for sub-region output at higher frequency.
util.f	Contains utility files such as interpolation and random number generation.
zefft.f	Computes the forward and inverse Fourier transforms of the rate terms.

2.4.1.1 *driver.f*

This module contains the main program that controls the entire simulation as well as four related subroutines and one function to control all simulation functions. The module primarily controls the execution of the program by calling the initialization modules, advancing time, calling time-marching subroutines, controlling the parameters related to time integration and available job time resources, and determining when to stop the execution.

1) <u>MAIN</u> - Controls the execution of the code and the output of information to files. The header contains the version history and the name of the current version. It also lists the files used and explains many of the parameters throughout TASS.

2) <u>DOIND</u> - Sets lateral boundary looping indices based on periodic or open boundary conditions.

3) <u>SECOND</u> - This function serves to extract the MPI system wall time for timing monitoring with the following subroutine <u>QTIMER</u>.

4) <u>QTIMER</u> - Accumulates, resets, and outputs information about the wall time used by different logical units contained within the TASS code (i.e., output, boundary conditions, initialization, microphysics).

5) <u>TIME2FILE</u> - This subroutine appends to the input character variable "file_name" a character representation of the TASS simulation time. This reduces the size of output files by creating a file name for each output time to aid with improved post-processing of TASS results.

6) <u>TIME_MONITOR</u> - This subroutine contains logic to query the TASS system environment to determine if it is running within the Portable Batch System (PBS) job scheduling environment. The subroutine then monitors PBS job time parameters to insure the simulation does not exceed the available job time limits. This subroutine relies on specific environment variables within the PBS environment as well as temporary hidden files to document operation progress for follow-on PBS jobs.

2.4.1.2 *advectl.f*

The primary purpose of this module is to compute the convective transport of terms other than momentum and pressure over the large time step. The module contains six subroutines and one function.

1) <u>GADVEC</u> - Transports the ground in the opposite direction of the grid movement to monitor accumulated precipitation substances at the ground.

2) <u>LADVEC_TVD</u> - Computes advection for scalar variables using Monotone Upstream-centered Scheme for Conservation Laws (MUSCL)-type scheme with Barth-Jesperson slope limiter (Ahmad and Proctor 2011) and diffusion using a 2nd order centered scheme.

3) <u>LADVEC</u> - Computes advection and diffusion for scalar variables using third-order Upstream- Biased Quadratic Interpolation (Leonard 1979).

4) <u>MARCHL</u> - Updates temperature, moisture and other scalar variables using one of the above subroutines in this module.

2.4.1.3 *advects.f*

This module is comprised of two coupled subroutines. The use of two rather than one subroutine was done to reduce compile times and to lessen issues a compiler may have with a large subroutine. The module computes the rate of change for velocity variables.

1) <u>MARCHS</u> - Computes advection and diffusion rates for the u and v components of velocity.

2) <u>ADVECTSA </u>- Computes advection and diffusion rates for the w component of velocity .

2.4.1.4 *allcomm.f*

This module contains all of the routine interfaces for MPI library calls. This subroutine was added by Elliott Schulman who reprogrammed the TASS code in 2004 to effectively utilize the MPI libraries. This module was added when migration from single-processor vector code to MPI-based vector code became advantageous, and serves to streamline MPI related functions from the original TASS vector code. While MPI calls are integrated through most of the code, TASS does not use the MPI libraries for parallel input or output (I/O). The master node, identified by the index of 0, controls all I/O functions. Any modifications to TASS require the user to review and carefully apply the MPI functions.

Global Reductions

The first group addresses the need to reduce information from the entire (global) domain. All nodes must call these routines and they are as follows.

1) <u>GLOBALSUM</u> - Computes the sum of a real scalar variable from all nodes.

2) <u>GLOBALSUM_K</u> - Computes the sum of a real vertical vector from all nodes at each horizontal level. This subroutine is not currently used, and will require user verification and modification before usage. Currently, the subroutine will abort a run and print a message to the output file.

3) <u>GLOBALSUM_IJ</u> - Computes the sum of a real horizontal two-dimensional array in each vertical column.

4) <u>GLOBALMAX</u> - Finds the maximum value of a real scalar of all nodes.

5) <u>GLOBALMIN</u> - Finds the minimum value of a real scalar of all nodes.

6) <u>GLOBALMAXLOC</u> - Finds the 3-D location (indexed in I, J, K) of the maximum value returned from GLOBALMAX.

7) <u>GLOBALMINLOC</u> - Finds the 3-D location in (indexed in I, J, K) of the minimum value returned from GLOBALMIN.

8) <u>GLOBALIMAX</u> - Same as GLOBALMAX but for an integer scalar.

9) <u>GLOBALSUMN</u> - Same as GLOBALSUM_K but for any orientation of a real vector of length N.

10) <u>GLOBALMAXN</u> - Compute the maximum values in a real vector of length N.

11) <u>GLOBALMINN</u> - Compute the minimum values in a real vector of length N.

<u>End-node Sends</u>

The next set of subroutines sends data from one end of the domain to the opposite end of the domain primarily for use with periodic boundary conditions. The first four subroutines are for real horizontal two-dimensional arrays, and the remaining six subroutines are for real three-dimensional arrays. Only boundary or edge nodes should call these routines

12) <u>QQYLINE_WEST</u> - copies a J-line located in the west-most node to the east most-node.

13) <u>QQYLINE_EAST</u> - copies a J-line located in the east-most node to the west most-node.

14) <u>QQXLINE_SOUTH</u> - copies an I-line located in the south-most node to the north-most node.

15) <u>QQXLINE_NORTH</u> - copies an I-line located in the north-most node to the south-most node.

16) <u>QQXPLANE_WEST</u> - copies a J, K-plane located in the west-most node to the east-most node.

17) <u>QQXPLANE_EAST</u> - copies a J, K-plane located in the east-most node to the west-most node.

18) <u>QQYPLANE_SOUTH</u> - copies an I, K-plane located in the south-most node to the north-most node.

19) <u>QQYPLANE_NORTH</u> - copies an I, K-plane located in the north-most node to the south-most node.

20) QQZPLANE_DOWN - Copies an I, J-plane located in the down-most node to the up-most node.

21) QQZPLANE_UP - Copies an I, J-plane located in the up-most node to the down-most node.

End-node Exchanges

The next set of subroutines serves to exchange boundary values. They all apply to three-dimensional real arrays used to filter boundary edges for periodic boundary conditions.

22) EWEXCHANGE - Copies J, K-plane data in east-most and west-most nodes to store in two temporary buffers.

23) NSEXCHANGE - Copies I, K-plane data in south-most and north-most nodes to store in two temporary buffers.

24) UDEXCHANGE - Copies I, J-plane data in down-most and up-most nodes to store in two temporary buffers. Currently not used.

25) QQZLINE_EW - Copies K-Line of data east-most and west-most nodes to store in two temporary buffers.

Gather / Scatter

The next set of routines is the workhorse of the MPI sending routines. They serve to store real data for sending in proper buffer locations (GATHER) or to read data from buffers back into array locations (SCATTER).

26) GATHER3D - Gathers data from the appropriate location within three-dimensional arrays into a temporary vector in preparation for node-to-node communication.

27) SCATTER3D - Scatters data from a temporary vector into the appropriate location within a three-dimensional array.

28) GATHER2D – Same as GATHER3D but for use with two-dimensional arrays.

29) SCATTER2D - Same as SCATTER3D but for use with two-dimensional arrays.

30) GATHER1D - Same as GATHER3D but for use with vectors.

31) SCATTER1D - Same as SCATTER3D but for use with vectors.

Generalized Edges

The next three subroutines send data from internal locations to halo or boundary edges of neighboring nodes. They can send 1, 2, or 3 points from each edge or all edges. In general, if a loop addresses a variable by an increment offset (i.e., I+1, J-1, K+3) the code should call either EDGE1 or ALLEDGES. If multiple offsets occur (i.e., (I+1, J+1), (I, J+2, K-1)) then use of the ALLEDGES subroutine also sets corner points necessary for proper neighboring processor addressing.

32) EDGE1 - Sends edge data to a neighboring node. The array can be either two- or three-dimensional, and the direction can be toward any one of the six neighboring nodes sharing a common face.

33) ALLEDGES - Sends edge data to all six neighboring nodes including corner points. The array can be either two- or three-dimensional. For three-dimensional arrays the subroutine can send either all six sides of the domain or only send the four horizontal edges.

34) JKEDGES - Sends edge data one point deep for a J, K-plane to all four neighboring nodes.

Share Planes

This sub-section addresses the need to conduct some computations in global mode, which requires reconstruction of a full two-dimensional plane from all of the processors identified by the planar location. Eventually, these subroutines should phase out as global operations disappear.

35) SHAREXPLANE - Creates a global two-dimensional array of a Y-Z plane at a specified I location. This subroutine is currently not used.

36) SHAREYPLANE - Creates a global two-dimensional array of an X-Z plane at a specified J location.

37) SHAREZPLANE - Creates a global two-dimensional array of an X-Y plane at a specified K location.

Broadcasts

The following subroutines broadcast data from one node to other nodes. These subroutines convey information computed in local nodes but required by other nodes. The last four subroutines communicate input data to all processors, since only the master node (0) reads input.

38) IJBROADCAST - Broadcasts the values of a real two-dimensional array from the down-most node to other nodes in each K column.

39) <u>IKBROADCAST</u> - Broadcasts the values of a real two-dimensional array from the south-most node to other nodes in each J column.

40) <u>JKBROADCAST</u> - Broadcasts the values of a real two-dimensional array from the west-most node to other nodes in each I column.

41) <u>ZBUILD</u> - Constructs a vertical vector of real data for the south- and west-most nodes.

42) <u>NBROADCAST</u> - Broadcasts a real vector from node 0 to all other nodes.

43) <u>NBROADCASTI</u> - Broadcasts an integer vector from node 0 to all other nodes.

44) <u>NBROADCASTL</u> - Broadcasts a logical vector from node 0 to all other nodes.

45) <u>NBROADCASTC</u> - Broadcasts a character vector from node 0 to all other nodes.

Start / Stop

The following two subroutines start and stop the MPI processes.

46) <u>MPISTART</u> - Initialized MPI.

47) <u>MPISTOP</u> - Finalize and stops MPI.

Other Functions

The next six subroutines do not fit into other categories, and they apply to legacy global operations and preparing data for output. The first three are subroutines for internal use by the share plane subroutines described above and other three subroutines contained in this subsection. The last three subroutines perform the function of gathering data from a specific level from a three-dimensional array or a two-dimensional array to a global two-dimensional array for output to disk.

48) <u>FINDIPROC</u> - Finds the specific node in the X direction that contains the desired I location. This subroutine is currently not used.

49) <u>FINDJPROC</u> - Finds the specific node in the Y direction that contains the desired J location.

50) <u>FINDKPROC</u> - Finds the specific node in the Z direction that contains the desired K location.

51) <u>COPYILEVEL</u> - Creates a global two-dimensional array comprised of either a local two-dimensional array or a specific I level of a three-dimensional array. This subroutine is currently not used.

52) <u>COPYJLEVEL</u> - Creates a global two-dimensional array comprised of either a local two-dimensional array or a specific J level of a three-dimensional array. This subroutine is currently not used.

53) <u>COPYKLEVEL</u> - Creates a global two-dimensional array comprised of either a local two-dimensional array or a specific K level of a three-dimensional array.

Global to Local and Local to Global

The first two subroutines in this subsection are similar to the gather/scatter routines mentioned above. The differences being that these routines are for subdividing global domain date or combining data to create the global domain arrays. They are for grouping data for output to disk, or splitting input data to each node. They can operate with both two- and three-dimensional arrays.

54) <u>GTOLX</u> - Sends real portions of data from a global array to the proper node.

55) <u>LTOGX</u> - Sends data from each node to reconstruct the global domain array.

56) <u>KMERGE</u> - Collapses the MPI array topology from three-dimensions to a two-dimensional *X-Y* array topology. The purpose of this subroutine is support legacy logic for fully-periodic wake vortex analysis.

2.4.1.5 *bounds.f*

This module addresses all issues dealing with the boundary conditions.

1) <u>BNDRYL</u> - Computes the lateral boundary conditions for scalar variables.

2) <u>BNDRYZ</u> - Computes the top and bottom conditions for scalar variables. The routine also specifies precipitation substances falling through the top boundary and computes terminal velocities at the ground for rain and hail. The user sets the specification of time history for precipitation substances.

3) <u>BUDGET</u> - Calculates the surface fluxes using a slab model for the surface energy budget.

4) <u>FILTER</u> - This subroutine filters potential temperature, vapor mixing ratio, *U, V,* and *W* at top and lateral boundaries.

5) <u>FILTER6</u> - This subroutine is a sixth-order space filter for arrays.

6) <u>FLUX</u> - Calculates surface heat flux, as directed from the boundary layer input file.

7) <u>GETFLX</u> - Returns the value of the surface heat flux or moisture flux.

8) <u>GETGEO</u> - Returns the value of the geostrophic wind and the current time.

9) HOLE - Periodically fills negative moisture values.

10) ORLP - Applies the Orlanski radiative boundary conditions to pressure along lateral boundaries.

11) ORLT - Applies the Orlanski radiative boundary conditions to potential temperature along lateral boundaries.

12) ORLUV - Calculates the Orlanski radiative boundary conditions to U and V velocities along lateral boundaries.

13) ORLW - Calculates the Orlanski radiative boundary conditions to W velocity along lateral boundaries.

14) PERTOPS - Calculates the vertical periodic boundary conditions for variables that are not periodic. The boundary condition is to maintain the same deviation from the environmental profile at the boundaries.

15) SPER - Sets the lateral periodic boundary conditions for scalar variables.

16) SPER2D - Sets the lateral periodic boundary conditions for scalar variables on a horizontal two-dimensional plane.

17) SPERV - Sets the vertical periodic boundary conditions for scalar variables (used only for Beltrami simulations).

18) VPER - Sets the lateral periodic boundary conditions for the velocity components. This subroutine also contains the specification of the upstream boundary for three-dimensional wake vortex simulations.

19) VPERV - Sets the vertical periodic boundary conditions for the velocity components.

20) PSH - A function used by subroutine BUDGET.

2.4.1.6 *dataproc.f*

This module prepares data to be written to a file. The input file controls the particular type and frequency of output.

1) FINDVEL - This subroutine is used for two-dimensional wake vortex simulations and it interpolates using Lagrange polynomials to determine the velocities.

2) FVEL3D - This subroutine is similar to FINDVEL but used for three-dimensional wake vortex simulations.

3) GRAPH - Prepares two- and three-dimensional data for output for later use in contour and vector plots. Some of the variables have options that are controlled within this subroutine.

4) PLOTF - Used for calculating the maximum 1-*km* averaged F-factor (Proctor et al., 2000). Also computes peak velocity change, ΔV, over any 4-*km* segment. This routine is only applicable for microburst simulations. Maps values as a function of height and time.

5) PLOTT - Determines the maximum and/or minimum of a variable at each horizontal plane. The variables evaluated are *W*, pressure, cloud water, cloud ice, rain, snow, hail, and total precipitation. In addition, it computes the average horizontal velocities and temperature. Maps values as a function of height and time.

6) PLOTZ - Determines the maximum or minimum of computed variables at each horizontal plane. The variables evaluated are radar reflectivity, vorticity, and temperature. Maps values as a function of height and time.

7) PLOTV - Computes circulation and position information for wake vortex simulations. This is only used for two-dimensional wake vortex simulations.

8) PLOTV3D - Computes the position information for three-dimensional wake vortex simulations.

9) TKEHIST - Computes the turbulent kinetic energy. For fully periodic domains this is done over the entire domain, otherwise it is done for a specific horizontal plane.

2.4.1.7 *decomp.f*

This module allows decomposition of the TASS domain into multiple subdomains for processing on-parallel processors. The first subroutine in this module reads the three-dimensional node configuration of the simulation domain, and then creates the required vectors and arrays to decompose the global domain to the three-dimensional node topology of the simulation. The remaining subroutines support the decomposition process.

1) DECOMP - This subroutine is the main driver for this module. It reads in node configuration from the control file (refer to section 3.1) and calls all the other subroutines of this module.

2) NEIGHBORS - Computes the node numbers for the six immediate neighbors for each node.

3) DIVIDE - Divides the domain among the processors by coordinate direction to insure no gaps or overlaps and includes halo regions.

4) POPULATE - Assigns coordinate ranges to each node.

5) BRINKPOINTS - Sets shorthand notation for location addressing of nodes at physical boundaries.

6) NODEPRINT - Prints the complete characterization of the decomposed domain. There are compile options in this routine to enable/disable output detail.

2.4.1.8 *diagvar.f*

This module calculates selective diagnostic variables needed during the time integration. The module contains the following subroutines:

1) <u>BUOY</u> - Calculates the density ratio term (buoyancy).

2) <u>DIFF</u> - Computes the subgrid eddy viscosity.

3) <u>DIV3D</u> - Calculates the three-dimensional anelastic divergence.

4) <u>DIV4</u> - Calculates three-dimensional divergence using 4th-order derivatives.

5) <u>FDIV</u> - Monitors diagnostic parameters and determines if the simulation is giving non-physical answers. Non-physical answers signal this subroutine to print out warning messages and terminate the run.

6) <u>RATE</u> - Calculates rainfall rate, hailfall rate, and accumulated precipitation at the ground surface.

7) <u>TEMP</u> - Calculates the temperature and the saturation-vapor mixing ratio.

2.4.1.9 *initial.f*

This module reads the input sounding and initializes the domain for the initial base state. The four routines in this module are called only at the beginning of a run.

1) <u>DATA</u> - Sets constants and initializes variables.

2) <u>FILTSET</u> - Sets the filter weight constants.

3) <u>IFILT</u> - Filters the input sounding.

4) <u>INITIAL</u> - Initializes the domain based on the input sounding. It also sets necessary atmospheric and microphysical parameters.

2.4.1.10 *initper.f*

This module, which is called by subroutine <u>INITIAL</u>, superimposes a perturbation field upon the initial base state, and completes initializations for desired phenomenology.

1) <u>INITPER</u> - Adds an initial perturbation to the domain. The input file controls the type of perturbation.

2) <u>ADDVORT</u> - Places a three-dimensional representation of a two-dimensional wake vortex in the domain. Can be called at initialization or at a later time during the simulation run. Therefore, is an entry into the subroutine <u>INITPER</u>.

3) <u>PINIT</u> - Calculates the initial pressure field based on an iterative inversion of Poisson's equation. Uses the initial flow and buoyancy fields.

4) <u>RINGVR</u> - Computes the radial velocity of a ring vortex.

5) <u>RINGVZ</u> - Computes the vertical velocity of a ring vortex.

6) <u>VELTAN</u> - Computes the tangential velocity for a two-dimensional vortex.

7) <u>VORTP</u> - Computes the pressure for an isolated two-dimensional vortex.

8) <u>ADVECT_INITIALIZE</u> - Defines a velocity field for use with advection testing.

2.4.1.11 *io.f*

This module contains the seven routines that perform input and output of information for diagnostic purposes. Section four contains the description of the contents output from these subroutines.

1) <u>HTREAD</u> - Reads in information from the boundary layer file relating to the surface heating.

2) <u>PLTPRO</u> - Writes information for planetary boundary layer simulation to FORTRAN unit 8.

3) <u>TAPE</u> - Writes to a three-dimensional data file (refer to section 4.1).

4) <u>TAPEF</u> - Writes time history profiles (refer to section 4.7).

5) <u>TAPER</u> - Writes to a two-dimensional plotting file (refer to section 4.5).

6) <u>TAPESIX</u> - Prints output that is an alphanumeric character representation of contour plots to the optional output file (refer to section 4.9).

7) <u>TAPET</u> - Writes time history profiles (refer to section 4.6).

2.4.1.12 *march.f*

This module computes parameters necessary for stable time advancement and advances the solution on the small time step.

1) <u>ADVMAX</u> - Computes the maximum velocity in the domain in order to check the numerical linear stability of the time step.

2) <u>NEWSTEP</u> - Computes the Adams-Bashforth time constants based on numerical stability criteria.

3) <u>TMCNS</u> - Initializes time dependent constants.

4) <u>TRACK</u> - Translates the grid to track desired phenomena.

5) <u>UWP</u> - Updates *U, V, W*, and *P* for each small time step.

2.4.1.13 *micro.f*

This module computes cloud microphysics based on bulk paramertization schemes, and includes over 60 cloud microphysics submodels. Most of the constants used in *micro.f* are set in subroutine INITIAL.

1) <u>MICPHY</u> - Computes cloud microphysics for rain, hail, snow, ice, and cloud water.

2) <u>MICPHYNOI</u> - This subroutine is similar to MICPHY but that it only computes warm microphysical relations (no ice processes).

3) <u>MICPHYC</u> - Computes only the cloud condensation and evaporation adjustments.

4) <u>WETGROUND</u> - Computes evaporation and cooling from a rain-soaked ground.

2.4.1.14 *restart.f*

This module handles all the input and output of restart data from TASS. There are two main subroutines <u>RSPL</u> and <u>WSPL</u>, which drive the I/O for the restart file. Due to the nonparallel I/O design of TASS, these routines use temporary arrays to construct global arrays prior to output or to read data. This module has backward compatibility for restart file formats from TASS version 9.05 onward. The program automatically adjusts to read the restart file format. As a convention, the first and second version numbers, currently 10.0, refers to a specific TASS restart file format. The second version number indicates code changes as in 2 of the current version 10.0.2.

5) <u>WSPL</u> - Main driver that writes the restart file to FORTRAN unit 2.

6) <u>RDSPL</u> - Main driver that reads the restart file from FORTRAN unit 3.

7) <u>RDARRAYS</u> - Reads the arrays from the restart file.

8) <u>RDSCALARS</u> - Reads the scalars from the restart file.

9) <u>WTARRAYS</u> - Writes the arrays to the restart file.

10) <u>WTSCALARS</u> - Writes the scalars to the restart file.

11) <u>WTARGS</u> - Writes arguments to the restart file.

12) <u>RDARGS</u> - Reads arguments from the restart file.

2.4.1.15 *sensor.f*

This module serves to obtain more frequent output of domain data over a sub-region of the global domain. The main driver subroutine of this module contains specific definition of the output subregion and output frequency. The module's name is derived from the need to create data sets for sensor technology development related to wake vortex simulations that did not require the full TASS domain data. This module contains use of the modern Fortran module program unit as a substitute for the common block data passing design of the main code. Unlike the rest of the code, this module contains direct calls to the MPI libraries to construct the necessary information to output the sub-region of data and writes output without calling the *io* module routines.

1) <u>SETUP_SENSOR</u> - Contains the sub-region, output frequency, and on/off control for the module. The sub-region definition relates to the global domain.

2) <u>GRAPH_SENSOR</u> - The main subroutine to create the global array of velocity and pressure for output via the subroutine SENSOR_TAPE.

3) <u>SENSOR_TAPE</u> - The main I/O driver for the module.

4) <u>ROI_LOC_TO_SENSOR</u> - Used by SENSOR_TAPE to populate the global array for output preparation.

2.4.1.16 *util.f*

This module contains a collection of routines that do not change and provide a variety of functions.

1) <u>RAN1</u> - Generates uniform random numbers between 0 and 1.

2) <u>RANDST</u> - Sets the random variation of a given value.

3) <u>SSCALE</u> - Used by subroutine <u>TAPESIX</u>.

4) <u>DIGPLT</u> - Heritage routine that simulates contour plots on a line printer.

5) <u>XDIGPLT</u> - Heritage routine that simulates contour plots on a line printer.

6) <u>STIUNI</u> - Interpolates sounding heights to grid point heights. This spline-fitting package was adapted from Cline (1974).

7) <u>QXZ057</u> - Part of the <u>STIUNI</u> interpolation package.

8) <u>QXZ058</u> - Part of the <u>STIUNI</u> interpolation package.

9) <u>QXZ060</u> - Part of the <u>STIUNI</u> interpolation package.

10) <u>QXZ061</u> - Part of the <u>STIUNI</u> interpolation package.

11) <u>QXZ062</u> - Part of the <u>STIUNI</u> interpolation package.

12) <u>QXZ063</u> - Part of the <u>STIUNI</u> interpolation package.

13) <u>QXZ110</u> - Part of the <u>STIUNI</u> interpolation package.

2.4.1.17 *zefftw.f*

This module contains logic to use the publicly available Fast Fourier Transform (FFT) library FFTW available at <u>www.fftw.org</u>. The module has logic to operate on either two-dimensional planes for laterally periodic boundary conditions or for three-dimensional arrays for fully-periodic boundary conditions. The module relies on the availability of the FFTW library being linked with the TASS executable and uses the MPI version of the FFTW library.

1) <u>EFFTW2D</u> - The main driver for the FFTW library for two-dimensional forcing. All of the necessary parameters to control the wavenumber, amplitude, and setup of the FFTW plan. The subroutine rejoins the individual U and V velocity components of the MPI domains into the global domain to re-assign the MPI processors to perform two-dimensional Fourier transform.

2) <u>EFFTW2DXEQ</u> - Performs the transformation to frequency space, adds energy to specific wavenumbers, and transforms the data back to the physical space on two-dimensional data.

3) <u>COPYTOSLABS2D</u> - Converts the real global velocity data to complex horizontal slabs for preparation for the Fourier transform.

4) <u>COPYFROMSLABS2D</u> - Reverts the complex horizontal slab array structure back to the real global three-dimensional array.

5) <u>EFFTW3D</u> - The main driver for the FFTW library for three-dimensional forcing. All of the necessary parameters to control the wavenumber, and amplitude. The subroutine rejoins the individual velocity components of the MPI domains into the global domain to re-assign the MPI processors to perform three-dimensional Fourier transform.

6) <u>SETUP3D</u> - Performs the setup of the forward and backward FFTW transform plans and outputs the structure of the data.

7) <u>EFFTW3DXEQ</u> - Performs the transformation to frequency space, adds energy to specific wavenumbers, and transforms the data back to the physical space on three-dimensional data.

8) <u>COPYTOSLABS3D</u> - Converts the real global velocity data to complex horizontal slabs for preparation for the Fourier transform.

9) <u>COPYFROMSLABS3D</u> - Revert the complex horizontal slab array structure back to the real global three-dimensional array.

10) <u>SETICNT2D</u> - Sets the two-dimensional array corresponding to the wavenumbers for energy addition based upon the value of the wavenumber limits defined in the main subroutine <u>EFFTW2D</u>. This routine also outputs the active wavenumber forcing information to the output file.

11) <u>SETICNT3D</u> - Sets the three-dimensional array corresponding to the wavenumbers for energy addition based upon the value of the wavenumber limits defined in the main subroutine <u>EFFTW3D</u>. This routine also outputs the active wavenumber forcing information to the output file.

3.0 INPUT FILES

There are three different files that control the execution of TASS and one file for restarting from the end of a previous simulation. The control file directs the basic operation of the simulation system, and the sounding file contains a meteorological sounding used for initialization of the base-state ambient variables. Both files are required to run TASS. An optional third file controls the planetary boundary layer information; however, if this file is not available, TASS disables the planetary boundary layer mode. A restart file generated at the end of each run allows for successive runs to continue without any loss of information. This allows a long running simulation to be broken into mutiple runs; thus, reducing the risk of losing a long-running program, as well as reducing the files size for data output of each run.

3.1　Control File

TASS reads the control file from the standard input or FORTRAN unit 5. This file uses a format in which each variable occupies specific locations on the line (Figure 3-1). The sample control file presented here represents a wake vortex simulation run. The part of each line to the right of the pound sign ("#") is for comments that give the name or a representative name of the variables in the TASS code. The explanation of this control file describes each variable as named within TASS for each line along with the formatting of each line.

Five subroutines read different parts of the control file. The subroutine <u>DECOMP</u> in module *decomp.f* reads the first line, and subroutine <u>MAIN</u> in module *driver.f* reads the next 9 lines of this file. Subroutine <u>DATA</u> in module *initial.f* reads the 11th line. Subroutine <u>INITIAL</u> in module *initial.f* reads lines 12 through 18, and subroutine <u>INITPER</u> in module *initper.f* reads the rest of the lines that control the simulation (19 to 24). The next to last variable in line 20 controls the optional additional vortex definitions beyond line 22. The data on lines beyond 22 can specify additional vortices, or comments to appear in the standard output file. Lines beyond those required by the above subroutines are comments, and subroutine <u>INITIAL</u> echoes them in the standard output file. Finally, all lines following the first line assume a fixed format. Each line is described below.

```
1)    1  16   9                                   # NIPROCS,NJPROCS,NKPROCS
2)     01.25    01.25    01.25 1.25    0.          # DX,DY,DZ,TMEND,TPRINT
3)   17500.0     1.0   1.000  .00E-4   0.2  00.040  # ZTOP,DZMAG,CHEN1,CHEN2,ZO,TPRINTC
4)      0.0 10000.0  2500.0  3000.0  5000.0 12000.0 # DTMP,RMAX,THGT,VRAD,VHGT,VTOP
5)   -0000.0   0000.   0000.  -0.000             # XCENTER,YCENTER,UGRID,VGRID
6)   .T. .T. .F. .T. .T.                      # GSKIP,BINARY,GTRACK,VORTON,DATOUT
7)     01.0     01.0    00.0    01.           # TIMEX,TIMEZ,TIMEF,TIMEV
8)   .T. .F. .T. .T.    0000   99215          # TEST,TRKCLD,RSPOOL,WSPOOL,IRSPL,IWSPL
9)   .F. .F. .F. .F. .T. .T.                  # E-W, N-S Outflow BC, PRECIPON,MICRON,DUSTON,ICEOFF
10)  .F. .F. .F. .T.                          # SUPERAD, DRYNEU, UNSTEADY, REINIT
11)    00.0     600.0     0.32     1.          # DLAT,ANC,DC,BUGZ ( IN SUB DATA )
12)     0.0    2500.   10000.   270.0          # AMVORT,ZVORT,TVORT, ANGSND
13)    000.0      45.    900.                  # BRMAX, TARRFT, HDNY
14)  .T. .T. .F. .F. .T.                       # UZERO,VZERO,VANL, VIRADB, PINITON
15)  .T. .F. .T. .F.    0.00     5.E4          # SKEWT,BNDYHMG,VELAVG,UVANL,TMPAD,DNOH
16)   2000.0      0.4    00.0      0.0          # UZTOP,UPOWER,UGRAD,VGRAD
17)  06 01 15 16 25 26 29 30 34 01 15 16 25 # ND,IDRANGE
18)  .T. .F. .F.   750.00     6.00             # SPHERE,VELIMP,TOPON,SIGH,QH (INITPER)
19)    -40.0 10000.0  2500.0     6.5           # SDIR,XMAX,YMAX (USED WHEN SPHERE=F),WPER
20)   1200.0      2.0  000.00  1  6            # FLXRAD,FLXMLT,VORTIME,VORTOT,INITYPE
21)  .F.  090.00     10.0 10000.0     5.0  # TWAVE,WDIR,WWMAX,XPLENGTH,HWAVZ
22)  .T.  565.00     3.75     022.     50.0  # HVORT,CIRC,RCORE,VALT,VSEP
23)    0.0 360.00      3.0    142.5     36.0      0.0 # TIME,CIRC,VCORE,VALT,VLAT,VSEP
24)                   # file datin.wake (v9.09)
```

Figure 3-1. An example of a TASS control file.

Line 1) This line contains information about the domain decomposition structure and the total number of CPUs used. The three integers on this line define the three-dimensional structure of the decomposed domain. Format: free format, integer values.

NIPROCS - The number of processors (sub-domains) in the coordinate direction cooresponding to I or *X*.

NJPROCS - The number of processors (sub-domains) in the coordinate direction cooresponding to J or *Y*.

NKPROCS - The number of processors (sub-domains) in the coordinate direction cooresponding to K or *Z*.

Hence, the full domain will be decomposed into a total number of subdomains equal to NIPROCS x NJPROCS x NKPROCS, requiring as many computer processors.

Line 2) This line contains information about the grid size and the end time for the run. Format: 6G8.0.

DX - Grid spacing in the *X* or east-west direction (*m*).

DY - Grid spacing in the *Y* or north-south direction (*m*). For two-dimensional simulations this variable is set to a large value (~1000 *m*).

DZ - Grid spacing in meters in the vertical direction (*m*). If this is a negative number then DZ is internally computed with the top level being at ZTOP, and the ratio of the grid size between the top and bottom being DZMAG. If CHEN1 _ 0 (line 2), then DZ represents Δz' in equation (104) of Proctor (1987).

TMEND - End time of simulation in minutes.

TPRINT - The beginning (first) time (*min*) for output of three-dimensional graph, and two-dimensional rates files. If this is negative then all output to these files is disabled and TPRINTC is not used. The content of the graph and rates files are explained later.

Line 3) This line contains supplementary information for the vertical grid and time to output information. Format: 6G8.0.

ZTOP - The height of the domain (*m*). Used only if DZ≤0 and in conjunction with DZMAG.

DZMAG - The ratio of the grid size at the top to that at the bottom. Used only if DZ ≤ 0 and in conjunction with ZTOP. Note that DZMAG should be a positive round number (e.g., 1.0, 2.0, 3.0 ...) (if DZMAG=1, no stretching occurs and DZ is constant). If DZMAG < 0 and DZ≤ 0, then stretching only applied between the ground and THGT (internally computes vertical grid size so that it is constant above THGT, and decreases below THGT such that it is reduced by a factor of 1/|DZMAG| at the ground).

CHEN1 - First stretching coefficient for vertical grid. For unstretched grid (DZ=constant) CHEN1=1.0. (not used if DZ<0).

CHEN2 - Second stretching coefficient for vertical grid. For unstretched grid (DZ=constant) CHEN2=0.0.

Z0 - Ground roughness parameter (*m*).

TPRINTC - Time interval (*min*) after TPRINT for output to three-dimensional graph and two-dimensional rate files. Information is appended to the file every TPRINTC minutes. If TPRINTC is 0, these files are only output at the end of the run.

Line 4) This line contains information to setup selected initial perturbations. Format: 6G8.0.

DTMP - Variable used to set up initial perturbation field.

DTMP<0: Initializes velocity field for a mesoscale vortex (with vertical axis). DTMP is the maximum tangential velocity (*m/s*) at the core radius VRAD.

DTMP=0: Off.

DTMP>0: Uses maximum temperature perturbation (^{o}C).

RMAX - Maximum horizontal radius (*m*), to apply the perturbation of either mesoscale vortex, velocity impulse, thermal impulse, or precipitation gush.

THGT - The height of the center of the thermal impulse with the peak at THGT/2 (*m*).

VRAD - Radius of the maximum tangential velocity in the mesoscale vortex (*m*).

VHGT - The height of the maximum tangential velocity in the mesoscale vortex or the top of the velocity impulse (*m*).

VTOP - Maximum height of the initial mesoscale vortex (*m*).

Line 5) The first two variables determine the horizontal location of the origin at initialization. If both are zero then the origin is the center of the domain. The other two variables specify the value to use for a fixed domain speed. Origin location defines center for initialization

of thermals, and other prescribed perburturbations that part of the initialization. Format: 4G8.0.

XCENTER - Initial horizontal *X*-distance from the center of the domain to the origin (*m*).

YCENTER - Initial horizontal *Y*-distance from the center of the domain to the origin (*m*).

UGRID - Horizontal east-west velocity of the translating grid (*m/s*).

> If UGRID > 500 then UGRID and VGRID are internally computed from the reference winds by integrating vertically from ground to the value contained in UGRID (*m*).

VGRID - Horizontal north-south velocity of the translating grid (*m/s*). (not used if UGRID greater than 500)

Line 6) Format: 5L4.

GSKIP - A logical variable that prevents the output of alpha-numeric character representation of contour line plots. Retained from early versions of TASS.

> True = Do not write (recommended).

> False = Write.

BINARY - A logical variable that controls the format of the three-dimensional graph file.

> True = Formatted ascii output.

> False = Binary, unformatted output (recommended).

GTRACK - A logical variable that controls the way the grid translates. Refer to subroutine TRACK in module *march.f* for details on the tracking logic.

> True = Translate the grid with the storm or wake vortex. The domain centering algorithm keeps the phenomena centered at the initial origin as determined from XSTART and YSTART.

> False = Translates with initial speed determined from UGRID and VGRID. If UGRID=VGRID=0, the domain is stationary.

VORTON - A logical variable that controls the type of domain tracking. This must be true to have the domain track with the lateral transport of wake vortices. Refer to subroutine TRACK in module *march.f* for details on the tracking logic.

> True = Translate the grid based on the vorticity of the cyclonic rotating storm or wake vortex.

> False = Translate the grid based on the centroid of the entire storm system.

DATOUT - A logical variable that enables the writing to the graph file.

> True = Write (recommended).

> False = Do not write.

Line 7) This line contains information to determine the frequency (in seconds) to output time history data. The format of the output files are described in the section 4.0. Zero values of these variables disable the output to the particular file. Refer to section 4.0 for further details regarding the format of these files. Format: 6G8.0.

TIMEX - The interval for time-height information generated by subroutine <u>PLOTT</u> in module *dataproc.f* (*s*).

TIMEZ - The interval for time-height information generated by subroutine <u>PLOTZ</u> in module *dataproc.f* (*s*).

TIMEF – If positive, the time interval for time-height information generated by subroutine <u>PLOTF</u> in module *dataproc.f* (*s*). If negative, the absolute value is the interval to compute the turbulence parameters for the domain (*s*).

TIMEV - The time interval for time-height information generated by <u>PLOTV</u> in module *dataproc.f* (*s*). If HVORT is false then this is automatically disabled.

Line 8) This line primarily contains information for restarting the simulation. Format: 4L4, 2I8.

TEST - A logical variable to test the initialization of the code.

True = Stop execution after initialization and output initial data to graph and rates files.

False = No effect. This is a common source of abrupt run termination due to the user forgetting to reset this to false for actual simulation after run checkout.

TRKCLD - A logical variable to control the translation of the domain for storm tracking.

True = Enables the use of cloud water in storm tracking.

False = Disables the use of cloud water in storm tracking.

RSPOOL - A logical variable to control the reading of a restart file.

True = Begin execution after reading from restart file.

False = Begin as new run.

WSPOOL - A logical variable to control the writing of a restart file. TASS will write restart information every IWSPL iterations and at end of simulation (time=TIMEND).

True = Enable the writing of a restart file (recommended).

False = Disable the writing of a restart file.

IRSPL - A positive integer representing an iteration number. If IRSPL is zero then the first restart time will be used from the restart file. Otherwise the first restart written at an iteration greater than IRSPL is used.

IWSPL - A positive integer that sets the number of iterations performed before writing out restart information. This variable is only effective if WSPOOL is true. Setting IWSPL very large will only generate a restart at the end of the simulation (recommended). If IWSPL is equal to zero it is ignored.

Line 9) Format: 5L4.

EOPEN - A logical variable that controls the handling of the East-West lateral boundaries. Although the user initially selects the boundary type based on this input variable, the actual bounary type may vary. The value of EOPEN initially sets the values of the East (EOPEN) and West (WOPEN) logicals.

True = Sets lateral boundaries to be open.

False = Sets lateral boundaries to be periodic.

SOPEN - A logical variable that controls the handling of the North-South lateral boundaries. Although the user initially selects the boundary type based on this input variable, the actual bounary type may vary. The value of SOPEN initially sets the values of the South (SOPEN) and North (NOPEN) logicals.

True = Sets lateral boundaries to be open.

False = Sets lateral boundaries to be periodic.

PRECIPON - A logical variable that controls the computation of precipitation substances.

True = Enables computation of precipitation substances.

False = Disables computation of precipitation substances.

MICRON - A logical variable that controls the computation of condensation variables.

True = Enables computation of condensation variables.

False = Disables computation of condensation variables and also sets PRECIPON to be false.

DUSTON - A logical variable that controls what is placed in the first domain of the dust array (ND=1).

True = Initializes the "DUST" to be a massless tracer.

False = Initializes the "DUST" to be insects (used in diagnostic output for Radar Reflectivity).

ICEOFF - A logical variable to disable ice phase physics.

True = Disables ice phase change. Ice subtances not allowed, water substances remain either vapor or liquid.

False = Enables computation of cloud ice, snow, and hail.

Line 10) This line contains logical variables to control the initialized environment. Format: 5L4.

SUPERAD - A logical variable that controls the admission of superadiabatic lapse rates.

True = Enables initialization with superadiabatic lapse rates.

False = Initialization procedure eliminates superadiabatic lapse rates from base state.

DRYNEU - A logical variable used to modify the input sounding.

True = Overrides sounding file and forces a dry neutral environment. This also forces PRECIPON, MICRON, and all boundary layer heating logical variables (UNHEAT, TSPEC, FLXSPEC, and EBUDG) to be false.

False = No effect.

UNSTEADY - A logical variable that effects the initialized field.

True = Initial base state is not balanced by residual terms.

False = Initial base state is balanced by residual terms and, therefore, is steady unless an initial impulse is added (recommended for open boundary simulations).

REINIT - A logical variable that effects the initialized field. Primarily used to rescale a resolved-scale turbulence field prior to injecting wake vortices.

True = Preserves the restart fields for pressure and velocity (minus the orginal base state) then adds them to the initial field generated by subroutine INITPER in module *initial.f*. The simulation time may be reset based upon the internal parameter TIMEZERO.

False = No restart file data is read in.

Line 11) Format: 4F8.0.

DLAT - Earth's latitude (degrees). This is used for determination of the coriolis forces. If DLAT=0, then the coriolis forces are disabled.

ANC - Number of condensation nuclei per cm^3 at an air density of 1 kg/m^3. Refer to Table 3-1 for example values.

DC - Dispersion coefficient for cloud droplet size distribution. Refer to Table 3 for example values.

BUGZ - The background (environmental) radar reflectivity due to insects in the atmospheric boundary layer (*dB*z). Recommended typical values are between -10 to 10 *dBz*

Table 3-1. Threshold Cloud Droplet Water Content (from Proctor 1987)

ANC	DC	Location	Reference
50	.366	Maritime	Simpson and Wiggert (1969)
689-927	.35-.38	Upwind of St. Louis	Fitzgerald and Spyers-Duran (1973)
1157-1472	.30-.32	Downwind of St. Louis	Fitzgerald and Spyers-Duran (1973)
2000	.146	Extreme Continental	Simpson and Wiggert (1969)
760-3166	.12-.32	Colorado High Plains	Knight et al. (1982)

Line 12) This line contains information about additional vorticity and orientation of model coordinates relative to initial data. Format: 4F8.0.

AMVORT - The relative vorticity of the environment (s^{-1}).

ZVORMX - The height of the peak environmental vorticity (*m*).

TVORT - The depth of the environmental vorticity (*m*).

ANGSND - The counter-clockwise rotation angle applied to the winds of the input sounding (degrees). Can be used to change the direction of the *X* and *Y* coordinates in TASS. Example: if ANGSND =90, winds along *X*-coordinate will be translated into the Y coordinate.

Line 13)   Format: 4F8.0.

BRMAX - The surface energy flux from solar radiation (*watts/m²*). Uses variables FLXRAD and FLXMLT (Line 19). Although still usable, this is an option from early versions of TASS that has been replaced by the planetary boundary layer option (see section 3.3).

TARRFT - Computes area (*m²*) as a function of altitude and time, where radar reflectivity is in excess of "TARRFT" (dBz). Used by subroutine PLOTZ in module *dataproc.f* to generate time-height histories.

HDNY - Density of hail (*kg/m³*).

Line 14)   This line contains information used to modify the input sounding. Format: 5L4.

UZERO - A logical variable to control the *U*-component of the sounding.

> True = Overwrites sounding input data with *U*=0 for initial base state.
>
> False = No effect.

VZERO - A logical variable to control the *V*-component of the sounding.

> True = Overwrites sounding input data with *V*=0 for initial base state.
>
> False = No effect.

VANL - A logical variable used to overwrite the initial *V*-component of the base state, as based on an analytical velocity profile as described in Figure 3-2, and only applies when UVANL is true, and both internal logicals UPOLY and VPOLY in subroutine INITIAL in module *initial.f* are false.

> True = Set the *V* profile to the analytical profile based upon the variables VGRAD, UZTOP, and UPOWER.
>
> False = No effect.

VIRADB – This is a logical variable to control the admissible lapse rate. If UNSTEADY is true this logical is treated as true.

True = Modify the initial temperature profile to eliminate lapse rates greater than the virtual adiabatic lapse rate - thereby, filtering out unstable layers of the sounding.

False = No effect.

PINITON – This is a logical variable to control the initialization of the pressure field.

> True = Balances the initial pressure, with the initial velocity, temperature, and humidity by an inversion of Poisson's equation (recommended). Minimizes slosh from unbalanced fields.
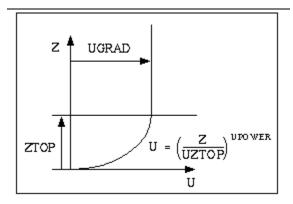
False = No effect.

Figure 3-2. Variation of velocity vs. height for the analytical velocity profile (z is altitude above the ground). The analytical profile is the same for both U and V with UGRAD being replaced by VGRAD for V.

Line 15)   Format: 4F8.0.

SKEWT - A logical variable enabling the output of a file containing the environmental data for the base state. File can be used for plotting of a sounding representing the initial base state. This option is automatically true if NEWSND is true (see Table 5-1).

> True = Output the base-state sounding to FORTRAN unit 23.

> False = No output of sounding data.

BNDYHMG - A logical variable controlling modification of the boundary layer temperature and humidity of the input sounding. Refer to Figure 3-3 for further explanation.

> True = Modify the input sounding for the given condensation temperature and pressure which are read in at the end of the input sounding file.

> False = No effect.

VELAVG - This is a logical variable used to smooth the initial, base-state velocity field.

> True = Smooth the velocity field.

> False = No effect.

UVANL - If true, this logical variable controls replacing of the input sounding with an analytical sounding based on the value of VANL and of internal logical variables UPOLY and VPOLY in subroutine INITIAL in module *initial.f*. The U-component of velocity is replaced with a constant value, UGRAD, above ZTOP, matched with an analytical profile below ZTOP, as described in figure 3-2. Similarly, the initial V-component is replaced with an analytical profile based on input for VGRAD only if VANL is true.

> True = If both UPOLY and VPOLY are false then the sounding assumes the form described in Figure 3-2 for U based upon the values of UGRAD, UZTOP, and UPOWER. However, if the value of VANL is true then it also controls the setting of V as described for the above for the parameter VANL. Otherwise, UPOLY and

VPOLY independently control over-writing the value of the U and V sounding components, respectively. Refer to the logic in subroutine <u>INITIAL</u> in module *initial.f* for further details.

False = No effect.

TMPAD – Adds "TMPAD" to the environmental base-state temperature at the first two grid points above the ground ($°C$). The user can use this variable to make lapse rates super-adiabatic near the ground.

NDOH - Intercept for hail used in size distribution (m^{-4}). Recommended values: ~4×10^4 for Great Plains storms and ~2×10^5 for southeastern U.S storms.
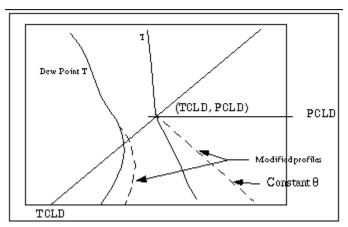

Figure 3-3. Sketch of skew-T plot describing modification to sounding with BNDYHMG set to true. Dashed lines represent the modified sounding based on input (`TCLD, PCLD`).

Line 16) This line contains variables to control the theoretical velocity profile. Refer to Figure 3-2 for details. Format: 4F8.0.

UZTOP - The maximum height of the theoretical velocity profile gradient (*m*).

UPOWER - This variable controls the power constant used in the theoretical profile.

UGRAD - The maximum *U*-component of the theoretical profile (*m/s*) assumed at the elevation UZTOP and above.

VGRAD - The maximum *V*-component of the theoretical profile (*m/s*) assumed at the elevation UZTOP and above.

Line 17) Variables used to initialize the environmental dust profile and three-dimensional DUST array for multiple dust domains. This line is only effective if the compiled parameter ND $\geq$ 4 (refer to section 5.1.1 for further details about ND). This option creates a horizontal layer of massless dust tracers. Format: 30I3.

NND - Number of dust domains to initialize, which must be no greater than the parameter ND described later in the internal parameters section.

IDRANGE - A pair of integers corresponding to each NND. Each pair refers to the lower and upper model K level to initialize each dust domain to 100 $g\ m^{-3}$.

Line 18)   These variables control how the initial perturbation is set up. Format: 3L4, 2F8.0.

SPHERE - A logical variable that controls the shape of the initial thermal impulse. This variable is only effective if DTMP is greater than 0.

True = Sets the thermal impulse to a spheroid.

False = Initializes a thermal trough along the direction SDIR and uses XMAX and YMAX (Line 19).

VELIMP - This is a logical variable that allows the initialization of a velocity impulse only if TOPON is false.

True = Initialize with a velocity impulse. Parameters to control the impulse are RMAX and VHGT (Line 4). This option can be used with thermal impulse.

False = No effect (recommended).

TOPON - This is a logical variable that enables a precipitation distribution to be initiated at the top boundary.

True = Enables precipitation to fall through the top boundary. Variables to control the precipitation are SIGH and QH. To control the time ramping of the precipi-tation during the run, refer to subroutine BNDRYZ in module *bounds.f* to specify precipitation time history. Note that FULLPER must be set to false.

False = No precipitation initialized and SIGH and QH are not used.

SIGH - Parameter that controls the spread of the distribution of precipitation on the top boundary.

QH - The maximum amplitude of the precipitation on the top boundary (g/m^3). Note: this QH is different from the QH in subroutine BNDRYZ).

Line 19)   These variables also control how the initial perturbation is set up. Format: 4F8.0.

SDIR - The orientation of the thermal trough in degrees. Zero degrees is in the Westerly direction.

XMAX - The maximum distance in the *X*-direction to extend the trough (*m*).

YMAX - The maximum distance in the *Y*-direction to extend the trough (*m*).

WPER - Amplitude of initial velocity perturbation (*m/s*).

Line 20)   The first two variables control the initial set-up of surface heating rate but are not currently used. The remaining variables deal with the two-dimensional wake vortex system configuration. Format: 4F8.0, I3.

FLXRAD - The maximum radius of the initial surface flux due to radiation (*m*). Currently disabled.

FLXMLT - A multiplier to the heating rate. Currently disabled.

VORTIME - The time to insert the first vortex system (*min*). Only effective if HVORT is true.

VORTOT - The total number of vortex systems to read. Only effective if HVORT is true, and it must be no larger than the parameter NVORMAX in the include file area22.h. Can be used to initialize wakes vortices from multiple aircraft.

INITYPE - The method to use to initialize the vortex system. Acceptable values are:

> 0 – Initialize a two-dimensional wake vortex pair in the x-z or vertical east-west plane.
>
> 1 – Initialize a three-dimensional ring vortex system.
>
> 2 – Initialize the three-dimensional domain with a two-dimensional vortex pair for all of the *Y-Z* or vertical north-south planes. This results in an open-ended three-dimensional vortex system originating from the eastward plane extending through the entire domain. When using this option, the value of UGRID should be positive to represent the forward speed of the generating aircraft.
>
> 3 – Initialize an upstream vortex forcing function for use when both the parameters WAKE3D in the include file param.h and VFORCE in the include file area22.h are true.
>
> 4 – Similar to option with the value of 2, but instead of initializing all Y-Z planes only initializes the first IPLANES (in subroutine <u>INITPER</u> in module *initial.f*, currently assumed as three) adding a half vortex ring closing the open vortex ends after IPLANES plane.
>
> 5 – This option only initializes a half vortex ring for use with the vortex forcing option, which requires both the parameters WAKE3D in the include file param.h and VFORCE in the include file area22.h to be true.
>
> 6 – Initialize the three-dimensional domain with a two-dimensional wake vortex pair for all of the *Y-Z* or vertical north-south planes.

The only methods that have been maintained are 0, 1, 2, and 6.

Line 21)   This line controls the setup of a horizontal gravity wave. Format: L4, 4F8.0.

TWAVE - A logical variable that turns on the initialization of a horzontal gravity wave.

WDIR - The orientation of the wave in degrees from the East direction.

WWMAX - The amplitude of the vertical velocity for the wave (*m/s*).

XPLENGTH - The length of the wave (*m*).

HWAVZ - The number of cycles within a length of the wave.

Line 22)   This line controls the set-up of a first vortex perturbation. Format: L4, 4F8.0, I2.

HVORT - This is the main logical variable that enables initialization of one or more wake vortex systems.

VCIRC - The value of the initial circulation for the first wake vortex system (m^2/s).

VRCORE - The value of wake vortex core radius (*m*). At minimum should be equal to 3 x grid size.

VALT - The initial height of the vortex system (*m*).

VSEP - This variable represents either the initial separation (b_o) of the wake vortex pair or the initial diameter of a ring vortex (*m*).

Line 23) This line and the next VORTOT-1 lines are of the same type. These lines control the parameters of the subsequently injected vortex systems. Each line corresponds to the next vortex system starting with the second system. This line is only effective if HVORT is true. Format: 5F8.0.

VORTIME - The time to insert the next vortex system (min). Can be inserted at time zero, or at a later time to simulate the interaction of wakes from an in-trail aircraft.

VCIRC - The value of the circulation for the wake vortex system (m^2/s).

VRCORE - The value of wake vortex core radius (*m*).

VALT - The initial height of the vortex system (*m*).

VLAT - The lateral position of the vortex system (*m*) relative to the first system ($\varDelta Y$).

VSEP - The initial separation (b_o) of the vortex pair (*m*).

Any remaining lines or comment lines are not used, but are written to the main output file.

3.2 Sounding Input File

The sounding input file currently has four format options. Two of the formats specify the altitude by distance above the ground, and the other two specify the altitude by the pressure height. An option also exists that allows specification of ambient cloud water content. Subroutine <u>INITIAL</u> in module *initial.f* reads the sounding file from FORTRAN unit 9 (Figure 3-4).

1)	13				
2)	1012.5	15.00	13.00	-1.8	-0.3
3)	1000.0	14.2	14.10	-1.80	-.30
4)	950.0	11.40	11.30	-1.20	-2.00
5)	900.0	8.60	3.00	-.50	-3.00
6)	850.0	5.50	0.00	.80	-3.20
7)	763.0	0.00	-5.00	1.14	-3.50
8)	628.0	-10.00	-20.00	1.35	-3.72
9)	500.0	-21.20	-31.70	2.10	-4.40
10)	400.0	-31.70	-51.70	2.00	-3.70
11)	298.0	-45.00	-65.00	.30	-2.40
12)	225.0	-56.7	-75.00	-.20	-3.70
13)	200.0	-56.70	-75.00	.00	-1.00
14)	100.0	-56.70	-75.00	-.90	1.40
15)	1				
16)	749.1	14.0			
17)					
18)	STANDARD ATMOSPHERE FOR TEMPERATURE -- THEORETICAL MOIST DEW POINTS --				
19)	AND DALLAS-FT WORTH, TX 0000 Z, 3 AUG 1985 WINDS				

Figure 3-4. An example of a sounding file.

Line 1) The first line contains a positive or negative integer, whose magnitude indicates the number of vertical points in the sounding. The value also controls the format of the remaining data.

N - The number of points in the sounding, along with information about the format of the sounding, subject to the following rules:

1. If 0<|N|<1000 then the number of points read is |N| and five columns of data are read per line.

2. If |N| is greater than 1000 then the number of points expected is |N|-1000 and six columns of data containing ambient cloud water is expected.

3. If N is less than 0 then the remaining data in the first column represents height above the ground (m).

4. If N is greater than zero, then the first column contains ambient pressure (*mb*).

Line 2) This line and the next N - 1 lines (2 through N + 1) are the same format. These lines contain the information of the sounding. Each line is a specific vertical sounding point indicating pressure (*mb*) or altitude (*m*), Temperature (^{o}C), dew point (^{o}C), *U*-component of wind (*m/s*) and *V*-component of wind (*m/s*). A sixth column for initial cloud water content (g/m^{3}) is needed if |N| > 1000.

Line N + 2) This line contains information about how to filter the sounding.

NFIL - The number of points to use in smoothing the raw sounding as follows:

=0   No filtering.

=1   Weak filtering (recommended).

=2 or 3   Three-point smoothing.

=4 or 5   Five-point smoothing.

38

=6 or 7   Seven-point smoothing.

NFIL=3 is recommended for short noisy soundings, while NFIL=5, or 7 for long noisy soundings. NFIL must be less than 8.

If altitude is in meters (N < 0) then the surface pressure (mb) must be included in the same line as NFIL.

Line N + 3) This line is needed if BNDYHMG is TRUE.

The two input variables within this line are:

PCLD - Pressure at the cloud condensation level (*mb*). (Figure 3-3)

TCLD - Temperature at the cloud condensation level (^{o}C). (Figure 3-3)

Lines N+4 and following) These lines contain comments about the sounding file and are echoed on the TASS main output file (FORTRAN unit 6).

3.3   Planetary Boundary Layer Input File

Subroutine <u>HTREAD</u> in module *io.f* reads the Planetary Boundary Layer (PBL) file from FORTRAN unit 7. If this file is not present, then the model will run in the original mode. The explanation for this file is expanded upon from Schowalter et al. (1996). Once used, subsequent restart runs must be able to read this file. The content of this file depends upon the options chosen and may vary greatly in form. To simplify the creation of this file, first read through the description of this file. Next, decide on the type of PBL simulation and obtain the salient information. Finally, assemble the information in the proper order in the file.

There are four parts that make up this file. The first part determines the type of PBL simulation and specifies the time-dependent PBL information. The second part sets the geostrophic wind. The third part determines the turbulent kinetic energy (TKE), and the fourth part stipulates supplementary information relating to dust and additional variables needed for the energy budget option.

The first part starts with a line containing six logical variables that control the type of PBL simulation. Only one of the first four variables may be true. If more than one of these is set to be true, the run terminates with an error message. If any of the first four variables are true, TASS introduces a random temperature perturbation into the first three layers of the domain to start up perturbations on a resolvable scale. The format of this line is 6L4.

UNHEAT - Specifies uniform heating. If true, a uniform heating is input at the first grid level.

TSPEC - Specifies temperature at a height above the ground. If true, the heat and moisture fluxes are calculated by assuming surface layer similarity. The Obukhov length is explicitly calculated for stress determination.

FLXSPEC - Specifies heat and moisture fluxes. The Obukhov length is explicitly calculated for stress determination.

EBUDG - Specifies an energy budget. The surface fluxes are calculated based on energy balance. This option requires additional information to be supplied in the fourth part of this file described later.

TKE - Specifies the turbulent kinetic energy (TKE) will be set at each grid level.

> True - Enable TKE specification.

> False - Disable TKE and the need for the third part of this input file.

FLXAD - Controls the Wyngaard scheme, which uses a surface flux conservation equation, derived by Wyngaard and Peltier.

> True - Enable surface flux conservation.

> False - Disable flux conservation.

If any one of the first four logical variables (UNHEAT, TSPEC, FLXSPEC, EBUDG) is true then the next line contains the number of data items for heating specification - XITMAX. The next XITMAX lines contain the heating data. This data is free format with each variable being separated by either a space or a comma. The data is dependent upon which logical variable is true as follows:

UNHEAT: time (min),   heat rate ($watt/m^2$)

TSPEC: time (min),   Temp at z_a ($°C$),   humidity at z_a (kg/kg)

FLXSPEC: time (min),   heat flux ($K•m/s$),   moisture flux (m/s)

EBUDG: time (min),   middle cloud fraction, high cloud fraction

If TSPEC is true then the next line after the heating data contains the value for z_a (m), which is the height above the ground.

The second part of the PBL file contains information for the geostrophic wind. The logical variable GFORCE on the first line enables the geostrophic wind (format L4). If this variable is true, the logical variable NOSTEADY in TASS should be set to be true; otherwise, NOSTEADY can be either true or false. If GFORCE is false, this completes this part of the PBL file, and further input will continue from part three. If GFORCE is true, the next line contains the logical variable

GWCONST, which controls the specification of the geostrophic wind (format L4). The remaining lines in part two contain the specification of the geostrophic wind in free format. Each line contains the U and then the V components of the geostrophic wind (m/s). The remainder of part two depends upon the value of GWCONST as follows:

> True: The next line contains the constant value to use

> False: The next KS-1 lines contain the profile of the geostrophic wind corresponding to a range from K=2 to K=KS. KS is a parameter in the include file param.h

TASS expects to read the third part of the PBL file only if TKE is true in the first part of this file. The integer on the first line, in free format, specifies the number of grid points used as a filter for the perturbation stream function. This variable (NSTR) must be odd and greater than 1. The next KS-3 lines contain the turbulent kinetic energy profile specification in m^2s^{-2} corresponding to a range from K=2 to K=KS-2.

The final part of this file begins with the logical variable DFLUX, which controls the introduction of dust throughout the simulation (format L4). If DFLUX is true, the next line is the value of the dust flux at the surface (DUSTIN in $kg\ m^{-2}s^{-1}$) in free format, and the initial domain does not contain any dust. The PBL file concludes at this point if EBUDG is false. Otherwise, the remainder of this section contains three lines with variables, in free format, necessary for the energy budget scheme as follows:

> 1) The first line contains information from which to start the simulation.

>> UTC - The Universal Coordinated Time at the initialization of the model.

>> DAY - The day of the year. A value between 1 and 365. This is used to compute the declination of the sun.

>> LNGT - The longitude of the domain (degrees).

>> TS - The surface temperature at initialization (K)

> 2) The next line contains information related to the moisture flux:

>> ALB - The ground albedo. A value from 0 to 1.

>> LAMS - The thermal conductivity of the soil (m^2/s).

>> TM - The substrate temperature (K).

>> WKW - The value of WG at which the atmosphere is saturated.

>> WMAX - The maximum value of WG, above which runoff occurs.

>> W2 - Soil moisture fraction at the substrate.

WG - Soil moisture fraction at the uppermost portion.

3) The final line contains:

CS - The heat capacity of the soil ($J{\cdot}Kg/(K{\cdot}m^3)$).

EMISS - The ground emissivity. A value from 0 to 1.

D1PRIME - The depth of the soil uppermost portion (*m*).

D2PRIME - The depth of the soil substrate (*m*).

An example heating input file is shown in Figure 3-5. The part of each line to the right of the pound sign "#" are comments giving the name of the variables in the TASS code. For this file, the first, second, and fourth parts are comprised of lines 1 through 14, 15 through 17, and 18, respectively. The logical TKE is false; so the third part is not present. This file is an example of a run with heat and moisture fluxes specified for a three-hour period. In addition, a constant geostrophic wind is specified, and there is not any dust flux.

```
 1)   .F. .F. .T. .F. .F.              # unheat, tspec, flxspec, ebudg,tke
 2)    12                       #if(.not.ebudg) xitmax: number of data items for heating
 3)   0.0       0.267427 0.00 3.00 0.00
 4)   10.26     0.18344 0.00 3.00 0.00
 5)   20.08     0.219194 0.00 3.00 0.00
 6)   32.07     0.189469 0.00 3.00 0.00
 7)   40.07     0.191247 0.00 3.00 0.00
 8)   50.17     0.180152 0.00 3.00 0.00
 9)   60.90     0.176366 0.00 3.00 0.00
10)   70.16     0.165216 0.00 3.00 0.00
11)   80.05     0.176542 0.00 3.00 0.00
12)   91.21     0.177656 0.00 3.00 0.00
13)   101.10    0.177599 0.00 3.00 0.00
14)   180.50    0.198946 0.00 3.00 0.00
15)   .T.                        #gforce:  if false, initial condition is in geo bal.
16)   .T.                        #gwconst:  if true geos. wind is constant with ht.
17)   3.0   0.0                  #ug, vg
18)   .F.                        #DFLUX
```

Figure 3-5. An example heating file.

3.4   Restart File

TASS reads from a restart file only if the input variable RSPOOL is true. TASS reads the restart file as an unformatted file from FORTRAN unit 3. For a restart run, the sounding file (FORTRAN unit 9) is not necessary. Restart files of all versions of TASS 9.09 are compatible.

4.0 OUTPUT FILES

TASS outputs data to several different files, which are unnamed FORTRAN unit files. On Unix machines the files default to the name "fort.#", where # is the unit number that created the file. Each output file also has an associated name. The following sections are in order of FORTRAN unit number and will explain the type, format, and purpose of the data in each file.

4.1   The Graph File (fort.1)

This file contains three-dimensional field data from a simulation. The output to this file is controlled by the input variables BINARY, DATOUT, TPRINT, and TPRINTC in the control file. The contents of this file are controlled by subroutine <u>GRAPH</u> in module dataproc.f, which the user must edit to switch between different options or to generate user defined output variables. The format of each variable written to this file is contained in subroutine <u>TAPE</u> in the module *io.f*. Due to the large volume of data output to this file, the format typically assumes an non-ASCII file format referred to as unformatted output dependent upon machine architecture. Assuming the logical BINARY is true, this format is:

```
 WRITE(IUNIT) IH,IWRT,IS,JS,KS,TIME,UGRID2,VGRID2,ORGX,ORGY,
1  DZ,CHEN1,CHEN2,DX,DY
 IF(IWRT.EQ.1) WRITE(IUNIT) A
```

where IUNIT is fixed at 1, and the output variables are defined as follows:

  IH - An integer that signifies which variable is to be output. This relationship is explained in Table 4-1.

  IWRT - An integer that assumes the value of either 0 or 1. Domain information is output if IWRT is 1.

  IS - The number of points in the *X*-direction of the domain for this simulation.

  JS - The number of points in the *Y*-direction of the domain for this simulation.

  KS - The number of points in the *Z*-direction of the domain for this simulation.

  TIME - The time of this output (seconds).

  UGRID2 - Two times the *X*-direction grid velocity (*m/s*).

  VGRID2 - Two times the *Y*-direction grid velocity (*m/s*).

  ORGX - The origin of the *X*-axis (*m*). (Changes with time due to grid translation)

  ORGY - The origin of the *Y*-axis (*m*). (Changes with time due to grid translation)

DZ, CHEN1, CHEN2 - These are mapping parameters to reconstruct the vertical grid spacing (m). They are defined in the input file.

DX, DY - The grid spacing in the x and y directions as defined in the input file.

A - The array of data to be written. Output of the array occurs only if IWRT is 1. IWRT is used to selectively turn off variables when their content is zero or not relevant to the current run. Since these are three-dimensional variables, it is easy to create very large graph files. Turning off unneeded variables helps to conserve disk space. The size of the A array is (NI,NJ,NK-1), and the order of output is the default FORTRAN ordering which is equivalent to:

```
WRITE(IUNIT) (((A(I,J,K),I=1,IS),J=1,JS),K=1,KS-1)
```

The explanation of each of the variables output and their relation to IH is shown in Table 4-1.

4.2   The Restart File (fort.2)

This machine dependent, unformatted file contains all of the variables required to seamlessly continue a simulation. The file is internal to TASS and has no purpose other than to enable subsequent runs of TASS to continue a simulation. This file must be moved to fort.3 for subsequent restart runs.

4.3   The Main Output File (fort.6)

This is the primary output file for a run. It contains the majority of the information about the run and states the parameters of the run along with the actual sounding used to initialize the domain. It also lists the type of perturbation initialized and contains information about key parameters of the run at every NCHECK large time steps. At the end of the file the central processing unit (CPU) usage for several subroutines is displayed.

4.4   The Planetary Boundary Layer Profile File (fort.8)

This file is abbreviated as the pro file. This file is an unformatted file that contains additional information, when used with the graph file, to compute PBL turbulence quantities, such as variance, covariance, and power spectra. It is written at the same model times as the graph file. The variables in this file are described in Table 4-2. The first four variables of this file are based on the grid height and are printed only once at the beginning of the file. The details of this file format are contained in subroutine PLTPRO in module *io.f.*

Table 4-1. Contents of Graph File

IH	Output Variable	Description
1	RLH	Relative Humidity with respect to liquid water (disabled).
2	U	X-component of wind velocity (*m/s*).
3	V	Y-component of wind velocity (*m/s*).
4	W	Vertical-component of wind velocity (*m/s*)
5	THETA	Temperature (°*C* or *K*), potential temperature (*K*), potential temperature deviation (*K*), or temperature deviation (°*C*).
6	P	Either pressure (*Pa*), or pressure deviation (*Pa* or *millibars*).
7	XIC	Liquid cloud water (kg/m^3).
8	XICI	Ice crystal water (kg/m^3).
9	CLD	Cloud water (kg/m^3). (sum of XIC + XICI + SNOW)
10	RRF	Radar reflectivity of precipitation (*dBz*).
11	XIV	Either water vapor (kg/m^3) or dewpoint temperature (°*C*) based upon the internal logical variable DEWOUT.
12	RRFI	Radar reflectivity of precipitation and insects (*dBz*).
13	EPOT	Equivalent potential temperature (°*C*).
14	RAIN	Rain water (kg/m^3).
15	SNOW	Snow water (kg/m^3).
16	HAIL	Hail water (kg/m^3).
17	DUST	Dust or insect concentration (*kg/kg*).
18	EDR	Subgrid eddy dissipation rate (m^2/s^3).
19	PRCP	Precipitable water (kg/m^3). (sum of RAIN + SNOW + HAIL)
20	RHO	Air density (k*g/m*^{3}).
21	TKE	Subgrid turbulent kinetic energy (m^2/s^2).
30	UDIF	U-component of the Beltrami velocity error (*m/s*).
31	VDIF	V-component of the Beltrami velocity error (*m/s*).
32	WDIF	W-component of the Beltrami velocity error (*m/s*).
33	PDIF	Beltrami pressure error (*Pa*).
≥101	SD##	Vortex tracer dust variable based on either vorticity or pressure as a function of the variable TRPRESS in subroutine <u>INITPER</u> (*kg/kg*). "##" is the number of the vortex system. For example, the value IH = 101 corresponds to SD01.

Table 4-2. Contents of Pro File

Output Variable	Description
ZH	Grid height at the cell center (m) printed at the first output only.
ZHH	Grid height at the cell face (m) printed at the first output only.
GM1	Grid mapping function for Z at the cell center (m^{-1}) printed at the first output only.
GMH1	Grid mapping function for Z at the cell face (m^{-1}) printed at the first output only.
ED	Subgrid eddy viscosity at the cell center (m^2/s).
EDZ	Subgrid eddy viscosity at the cell face (m^2/s).
RHO	Density (kg/m^3).
QFLX	Heat flux ($K\,m/s$).
TFLX	Temperature flux ($K\,m/s$).
STSU	Surface stress in the U component direction ($kg/(ms)^2$).
STSV	Surface stress in the V component direction ($kg/(ms)^2$).

4.5   The Rates File (fort.11)

This file contains horizontal slices of the domain for selected variables. Most of the variables are at the surface and include diagnostic fields for precipitation rates. This file is written with the same time interval as the graph file. Setup of the data for output to this file is done in subroutine GRAPH in module dataproc.f. The current format for each variable that is written to this file is contained in subroutine TAPER in the module io.f and is as follows:

```
     WRITE(IUNIT,100) IS,JS,IH
     WRITE(IUNIT,101) TIME,ORGX,ORGY,DX,DY,A
 100     FORMAT (3I5)
 101  FORMAT(8E12.4)
```

The description of the output parameters is the same as for the graph file with the array "A" being two- and not three-dimensional. The size of the array is IS by JS data points and is written in the default FORTRAN ordering which is equivalent to:

```
       WRITE(IUNIT) ((A(I,J),I=1,IS),J=1,JS)
```

The description of the variables contained in this file is given in Table 4-3.

Table 4-3. Contents of Rates File

IH	Output Variable	Description
4	W	*W*-component of the velocity at 1 *km* above the surface (*m/s*).
6	P	Pressure at the surface (*Pa*).
10	RRF	Radar reflectivity at 2 *km* above the surface (*dBz*).
21	RFRATE	Rain fall rate (*inches*/*hour*).
22	HFRATE	Hail fall rate (*inches*/*hour*).
23	TPREC	Total precipitation (*inches*).
24	THAIL	Total hail (*inches*).
25	VAPOR	Water vapor at the surface (kg/m^3).
28	THETA	The potential temperature at the surface (*K*).
29	VORT	Vertical vorticity just above the surface (s^{-1}).
30	SPEED	Horizontal wind speed at the surface (*m/s*).
31	DUST	Dust at the surface.
32	VIP	Vertically integrated precipitation (*kg*).

4.6   The Time Height for Cloud (thc) File (fort.12)

This file is abbreviated as the thc file. In addition to cloud variables, this file also includes information about the vertical velocity and pressure. The variable TIMEX in the input controls the frequency of writing to this file—a vector of vertical data is written every TIMEX seconds. Each point in this vector is either the maximum or minimum of the horizontal plane of the corresponding three-dimensional array. The data is set up in subroutine PLOTT in module dataproc.f, and subroutine TAPET in module io.f controls the actual writing of data to this file. The data is written with the following format:

```
      WRITE(IUNIT,100) IH,KS1,NN
      WRITE(IUNIT,101) TIME,DZ,CHEN1,CHEN2,A
 100  FORMAT(3I5)
 101  FORMAT(8E12.4)
```

The description of the output parameters is the same as that for the graph file with the exception of the size of the array, "A". The variable NN is not used. The array "A" is a vector with a length of KS-1. The description of the variables contained in this file is given in Table 4-4.

Table 4-4. Contents of thc File

IH	Output Variable	Description
1	WMAX	Maximum vertical-component of the wind (m/s) for each horizontal plane.
2	WMIN	Minimum vertical-component of the wind (m/s) for each horizontal plane.
5	PMAX	Maximum pressure (pa) for each horizontal plane.
6	PMIN	Minimum pressure (pa) for each horizontal plane.
7	CWMAX	Maximum cloud water (kg/m^3) for each horizontal plane.
8	CIMAX	Maximum ice crystal water (kg/m^3) for each horizontal plane.
9	RNMAX	Maximum rainwater (kg/m^3) for each horizontal plane.
10	SNMAX	Maximum snow water (kg/m^3) for each horizontal plane.
11	HLMAX	Maximum hail water (kg/m^3) for each horizontal plane.
22	XPMAX	Maximum total precipitation (RAIN+SNOW+HAIL) (kg/kg).
30	UVAVE	Average U-velocity component (m/s).
31	VAVE	Average V-velocity component (m/s).
32	TAVE	Average potential temperature component (K).

4.7   The Time Height for F-factor (thf) File (fort.13)

This file is abbreviated as the thf file. This file contains the time history of the F-factor (Bowles 1990) and peak horizontal wind change at low levels. TIMEF in the input file controls the frequency of writing to this file. The setup is accomplished in subroutine PLOTF in module dataproc.f and the output is performed in subroutine TAPEF in module io.f. The format of the output file is:

```
    WRITE (IUNIT,100) IH,NN
    WRITE (IUNIT,101) TIME,DZ,CHEN1,CHEN2,A
100  FORMAT(2I5)
101  FORMAT(8E12.4)
```

where the description of the variables is the same as the thc file, with NN being the length of the vector A. NN corresponds to the highest vertical level that is below 2000 meters above the ground, since the windshear is not considered a hazard below this altitude. Computation of the F-factor assumes level flight paths and a flight speed of 75 m/s. The description of the data in this file is in Table 4-5.

Table 4-5. Contents of thf File

IH	Output Variable	Description
18	DVEW	Maximum East-West wind change along any 4 *km* segment (*m/s*).
19	DVNS	Minimum North-South wind change along any 4 *km* segment (*m/s*).
20	FMXEW	Maximum 1-*km* averaged F-factor in the East-West direction.
21	FMXNS	Maximum 1-*km* averaged F-factor in the North-South direction.

For a Beltrami flow test simulation, the contents of the above file are replaced with Beltrami information as described in Table 4-6. In this mode, output is written to this file every NCHECK large time steps. The setup and output of information is in subroutine FDIV in module *diagvar.f*. The format of the output is:

```
WRITE(13,'(10(G15.7,1X))') TIME,TOTKE,PKERROR,UMAX,PUERROR,VMAX,
1  PVERROR,WMAX,PWERROR,PPERROR
```

Table 4-6. Contents of thf File for Beltrami Flow Simulation Mode

Output Variable	Description
TIME	Time of output (*s*).
TOTKE	The total kinetic energy of the TASS domain (m^2/s^2).
PKERROR	RMS error of TOTKE.
UMAX	The maximum *U* in the domain (*m/s*).
PUERROR	RMS error of UMAX.
VMAX	The maximum *V* in the domain (*m/s*).
PVERROR	RMS error of VMAX.
WMAX	The maximum *W* in the domain (*m/s*).
PWERROR	RMS error of WMAX.
PPERROR	RMS error of the maximum pressure in the domain.

This file contains error information for the root-mean-square (RMS) error. The following relationship gives the RMS error:

$$Error_{RMS} = \sqrt{\frac{\displaystyle\sum_{i,j,k}(Q_{MODEL} - Q_{ANL})^2}{\displaystyle\sum_{i,j,k}Q_{ANL}^2}} \; ,$$

where $\displaystyle\sum_{i,j,k}$ is the sum over the entire domain, and Q is a velocity component, pressure or kinetic energy.

4.8   The Time Height for Reflectivity (thz) File (fort.14)

This file uses the same format as the thc file and is abbreviated thz. The file contents also include information about vorticity and temperature. The setup of the data is performed in subroutine <u>PLOTZ</u> in module *dataproc.f*, and the frequency of output is controlled by the variable TIMEZ. The contents of this file are described in Table 4-7.

Table 4-7. Contents of thz File

IH	Output Variable	Description
3	RFMAX	Maximum radar reflectivity (*dBz*) for each horizontal plane.
12	ZAREA	The area in the plane that exceeds the radar reflectivity value TARRFT (m^2).
4	VORTMX	Maximum vorticity (s^{-1}).
13	VORTMN	Minimum vorticity (s^{-1}).
16	TMAX	Maximum temperature (*K*).
17	TMIN	Minimum temperature (*K*).

The vertical component of vorticity is used for VORTMX and VORTMN when HVORT is false. When HVORT is true, the vorticity component is dependent upon WAKE3D. The y component of vorticity is used when WAKE3D is false, and the x component is used when WAKE3D is true.

4.9   The Optional Output (opt) File (fort.15)

The input variable GSKIP controls output to this file. When this variable is true, TASS outputs an alphanumeric character representation of contour plots to this file. The logic to control

this output is by calls to the subroutine <u>TAPESIX</u> in the module io.f. The calls to <u>TAPESIX</u> exist in the subroutine <u>GRAPH</u> in the dataproc.f module. Although it still exists, it is obsolete.

4.10 Position File for Port Vortex (thvp) (fort.16)

This file adopts the name as the thvp file and applies only to wake vortex or Beltrami runs. This file contains data about the vortex with positive vorticity or the left vortex, and for Beltrami runs it contains the error analysis parameters.

For wake vortex simulations, this file contains vortex positions and circulations, which are updated every TIMEV seconds. The circulation values based on the integration of the X- or Y-component of vorticity for either case of three- or two-dimensional wake runs. For the three-dimensional wake runs, the calculated circulations may become less accurate, since the vortex may no longer align with the X-axis due to twisting and looping. More accurate and extensive analysis of three-dimensional wake runs requires post-processing programs.

For two-dimensional wake runs, this file contains average and total circulations values. Average circulation assumes the following relationship:

$$\Gamma_{a,b} = \frac{\int_a^b \Gamma_r dr}{\int_a^b dr},$$

where a and b are the radii of the averaging interval, and Γ_r is the total circulation over a radius of r. The relation for Γ_r is given by:

$$\Gamma_r = \iint_r \xi \, dA,$$

where ζ is the Y component of vorticity. Also, Γ_r represents the total circulation at a specific radius. If the vortex gets closer to the boundary than the maximum radius needed for the circulation computation, or a valid vortex signature no longer exists, a value of -9999 appears in the file. The output of this file changes depending upon the dimensions of the simulation. For two-dimensional simulations, the output contains the position, interpolated velocities, and selected circulation values. For three-dimensional simulations, the output omits the circulation values and prints vortex parameters for each Y-Z plane followed by an average of all planes of the domain. The two subroutines <u>PLOTV</u> and <u>PLOTV3D</u>, both in the module *dataproc.f*, control the output to this file for

two- and three-dimensional simulations, respectively. The user can select different output options by editing the appropriate subroutine.

The vortex output for two-dimensional simulations from the subroutine <u>PLOTV</u> varies based on key logical and array values. The user can select to have a concise or elaborate output for the vortex pair. The logical variable SHORTOUT controls the detail of the output. If SHORTOUT is true only a select number of averaged circulations and non-averaged (total) circulations appear in the output file.

The array ARADIUS (averaged) and the vector RADIUS (total) controls the type of circulation output (e.g, 5-15m averaged circulation, total circulation at 15m radius). Currently, the number of specific averaged circulations is 7, and the number of total circulations is 5 for each port and starboard vortex. The user can change this by altering the values of the parameters NUMACIRC and NUMCIRC for averaged and total circulations, respectively. To set the radii to compute the circulation over, the user must change the values of ARADIUS and/or RADIUS. For averaged circulation values, ARADIUS is an array of rank 2 with the first rank dimensioned by NUMACIRC and the second rank set to 2. The second rank controls the range, in meters, of the average circulation calculations. The first value is the inner or smaller radius and the second is the outer or larger radius. For total circulation values, the vector RADIUS, dimensioned by NUMCIRC, sets the radii for computation of total circulation in the output. Currently, the code fixes the first 5 averaged and the first 3 total circulations to fixed values. The remaining circulations depend upon the vortex pair separation. The user must examine the code to see the details of these variables and how to alter them to achieve the desired output. Table 4-8 contains an overview of the two-line output at each output time when the logical variable SHORTOUT is true.

When the logical SHORTOUT is false, the output changes to include the output of the vortex circulation radial profile. Three variables, two of which appear in the output, control the profile definition. They define the starting radius (RADSTART), the maximum radius (AVERADMAX), and the incremental radius (DELRAD). The value for AVERADMAX assumes the larger of the initial vortex separation or the current vortex separation. The user can alter the values of RADSTART and DELRAD, but changing these values will alter all averaged circulation computations. The output changes to become a three-line output for each output time. Table 4-9 contains the description of the output when SHORTOUT is false.

Whatever the value of SHORTOUT, the user will need to construct post-processing software to analyze the output.

Table 4-8. Contents of the thvp file for two-dimensional vortices when SHORTOUT is true.

Line Number	Output Variable	Description
1	TIME	Time of output (s).
1	XPOS	The horizontal position (cross-track position) of the vortex (m).
1	ZPOS	The vertical position of the vortex (m).
1	UVEL	The interpolated component of the horizontal velocity (m/s).
1	WVEL	The interpolated component of the vertical velocity (m/s).
1	RADMAX	The radius at with the circulation profile is the largest (m).
1	CIRCMAX	The value of circulation at the radius RADMAX (m^2/s).
1	CSEP	The cross-track separation of the vortex pair (m).
1	IVORT	An integer referring to the vortex system. Not written if only one vortex system is active in the simulation.
2	ACIRC	The first NUMACIRC values on this line are the averaged circulations as defined in subroutine PLOTV (m^2/s).
2	CIRC	The remaining NUMCIRC values on this line are the total circulations as defined in subroutine PLOTV (m^2/s).

Table 4-9. Contents of the thvp file for two-dimensional vortices when SHORTOUT is false.

Line Number	Output Variable	Description
1	NAVETOT	The number of circulation values that comprise the circulation profile.
1	NUMACIRC	The number of averaged circulation values output on line 2
1	NUMCIRC	The number of total circulation values output on line 2.
1	TIME	Time of output (s).
1	XPOS	The horizontal position (cross-track position) of the vortex (m).
1	ZPOS	The vertical position of the vortex (m).
1	UVEL	The interpolated component of the horizontal velocity (m/s).
1	WVEL	The interpolated component of the vertical velocity (m/s).
1	RADMAX	The radius at with the circulation profile is the largest (m).
1	CIRCMAX	The value of circulation at the radius RADMAX (m^2/s).
1	CSEP	The cross-track separation of the vortex pair (m).
1	RADSTART	The radius at which the circulation profile starts (m).
1	DELRAD	The radial increment for each point of the circulation profile (m).
1	IVORT	An integer referring to the vortex system. Not written if only one vortex system is active in the simulation.
2	ACIRC	The first NUMACIRC values on this line are the averaged circulations as defined in subroutine PLOTV (m^2/s).
2	CIRC	The remaining NUMCIRC values on this line are the total circulations as defined in subroutine PLOTV (m^2/s).
3	CIRCRAD	There are NAVETOT values on this line that define the values of the profile (m^2/s).

For three-dimensional vortex simulation, the form of the output changes by deleting the circulation information and computing the vortex position at each I plane of the domain. The subroutine PLOTV3D controls the output in this case. Table 4-10 describes the contents of this file in this mode (in the table, IS4 is shorthand for IS-4, similarly for IS3 and IS2).

Table 4-10. Contents of the thvp file for three-dimensional vortices.

Line Number	Output Variable	Description
1 to IS4	I	I station of this line of output.
1 to IS4	YPOS	The horizontal position (cross-track position) of the vortex (*m*).
1 to IS4	ZPOS	The vertical position of the vortex (*m*).
1 to IS4	VVEL	The interpolated component of the horizontal velocity (*m/s*).
1 to IS4	WVEL	The interpolated component of the vertical velocity (*m/s*).
1 to IS4	IVORT	An integer referring to the vortex system. Not written if only one vortex system is active in the simulation.
IS3	TIME	Time of output (*s*).
IS3	AVEYPOS	The domain average horizontal position (cross-track position) of the vortex (*m*).
IS3	AVEZPOS	The domain average vertical position of the vortex (*m*).
IS3	AVEVVEL	The domain average interpolated component of the horizontal velocity (*m/s*).
IS3	AVEWVEL	The domain average interpolated component of the vertical velocity (*m/s*).
IS3	AVESEP	The remaining NUMCIRC values on this line are the total circulations as defined in subroutine PLOTV (m^2/s).
IS3	IVORT	An integer referring to the vortex system. Not written if only one vortex system is active in the simulation.
IS2	TIME	Time of output (*s*).
IS2	CROWFACT	The vortex linking factor as defined by the following equation: $$CROWFACT = \frac{SEPMAX - SEPMIN}{SEPMAX + SEPMIN}$$
IS2	SEPMAX	The maximum cross-track separation (*m*).
IS2	SEPMIN	The minimum cross-track separation (*m*).
IS2	MAXLOC	The I plane where SEPMAX occurs.
IS2	MINLOC	The I plane where SEPMIN occurs.

For Beltrami mode simulations, the file content changes to the variables explained in Table 4-11. The subroutine FDIV in module *diagvar.f* computes and writes this data every NCHECK large time steps. The error in this file is the local percent as computed by the following relation:

$$Error_{Percent} = Max_{i,j,k} \left| \left(\frac{(Q_{MODEL} - Q_{ANL})}{Q_{ANL}} \right) \right| 100 \, ,$$

where $Max_{i,j,k}$ is the maximum over the entire domain, and Q is a velocity component, kinetic energy, or pressure.

Table 4-11. Contents of the thvp File for Beltrami Flow Simulation Mode

Output Variable	Description
TIME	Time of output (s).
TOTKE	The total kinetic energy of the TASS domain ($m^2 s^{-2}$).
XPERROR	The maximum absolute local percent error of TOTKE.
UMAX	The maximum U in the domain (*m/s*).
PUERROR	The maximum absolute local percent error of UMAX.
VMAX	The maximum V in the domain (*m/s*).
PVERROR	The maximum absolute local percent error of VMAX.
WMAX	The maximum W in the domain (*m/s*).
PWERROR	The maximum absolute local percent error of WMAX.
PPERROR	The maximum absolute local percent error of pressure.

4.11 Temporary Timing File (fort.17)

This file is a temporary file used to contain timing data that is written at the end of the main output file. It has no useful data other than internal TASS recordkeeping of run time data.

4.12 Position File for Starboard Vortex (thvn) (fort.18)

The information in this file is for the vortex with negative vorticity, or the right vortex; the contents are the same as the thvp file (fort.16).

For Beltrami simulations, this file contains the same variables as that of Table 4-11. The error is the global percent error as defined by the following equation:

$$Error_{Percent} = \left(\frac{Max_{i,j,k}\left(|Q|_{MODEL}\right) - Max_{i,j,k}\left(|Q|_{ANL}\right)}{Max_{i,j,k}\left(|Q|_{ANL}\right)} \right) 100 \, ,$$

where $Max_{i,j,k}$ is the maximum over the entire domain, and Q is a velocity component, kinetic energy, or pressure.

4.13 The Time Height for Boundary Layer (thbl) File (fort.19)

The format of this file is the same as the thc file, and fort.19 is referred to as the thbl file. This file contains horizontally averaged wind and temperature values at each level. Table 4-12 lists the contents of this file.

Table 4-12. Contents of the thbl File

IH	Output Variable	Description
30	UAVE	Average east-west velocity (*m/s*).
31	VAVE	Average north-south velocity (*m/s*).
32	TAVE	Average potential temperature (*K*).

4.14 The Sounding File (fort.23)

This file contains the processed environmental sounding generated from the TASS initial profile data based upon the internal logical variables PHGT and NEWSND. When NEWSND is true, TASS outputs the sounding and then stops, which serves to generate sounding files for use with wake vortex runs based on values of thermal stratification and input wake vortex parameters. The variable PHGT controls the type of height data reference contained in the file. If PHGT is true, the data in the file is in pressure based heights; otherwise, the data heights are physical heights. The subroutine INITIAL in module *initial.f* generates this file, which contains data for each model level. The format is similar to the fort.9 file. When NEWSND is true, the last lines of this file contain the data used to generate the sounding file.

Line 1) The first line contains an integer of the number of vertical points in the sounding. When this number assumes a negative value the heights are physical elevations, which only can occur when NEWSND is true.

　　N - Number of sounding points.

Line 2) This line contains the pressure and temperature at the surface if the heights are pressure based. Otherwise, for physical height data this data has no relevance so it does not appear in the file. If present, the variables are:

　　PSR - Atmospheric pressure (*mb*).

　　TMPR - Temperature (°*C*).

　　DWPTR -Dewpoint temperature (°*C*).

　　U Velocity set to 0 for format consistency (*m/s*).

　　V Velocity set to 0 for format consistency (*m/s*).

Line 3) This line through N+1 are of the same format. These lines add the altitude information if based on pressure heights. Each line is a specific vertical sounding point corresponding to the vertical levels of the domain. The values in each column along the line represent:

PSR - Atmospheric pressure (*mb*) or physical height (*m*) depending on the PHGT logical.

TMPR - Temperature (°*C*).

DWPTR - Dewpoint temperature (°*C*).

U0R - East-West component of the wind (*m/s*).

V0R - North-South component of the wind (*m/s*).

ZPR - Altitude (*m*) above the ground for a pressure based height file.

5.0 INTERNAL TASS PARAMETERS

TASS utilizes numerous parameters throughout the program. Parameters internal to TASS reside in two locations: in the include files, and within the modules. The parameters within the include files will be explained first, followed by those within the modules.

5.1   Parameters in the Include Files

5.1.1   <u>Parameters in the include file param.h</u>

This file contains the parameters to define the sizes of the global and local computational domains and to determine the fundamental mode of the simulation system. The param.h file appears in all of the modules and can disable portions of the code for compiler optimization and feature flexibility. Figure 5-1 contains an example of this file for which the description of its variables follows.

Line 1)   The first three parameters control the size of the physical computational domain and include boundary points. NI, NJ, and NK are the sizes of the x, y, and z directions, respectively.

Line 2)   The three parameters on this line set the lower limit of the number of processors required to conduct the simulation. The purpose of these is to set the individual node data arrays to reduce the size of the TASS executable. Setting these values to one does not prevent running with many MPI processors and signifies the node data arrays to be the same size as the global data arrays. However, it is recommended that these values be set to the same values used for NIPROCS, NJPROCS, and NKPROCS as used in the control file.

Lines 4-6)   These three lines set the limits of the local node array sizes based on dynamic parameter assignment based on the processor limits defined above.

Line 7)   The parameter NG determines the maximum number of nodes a single simulation can handle. The important parameter is NG1, because nodes increment from 0. The parameter NGUTTER controls the size of the halos of node edges for inter-process boundary communications. This should not change, and can never be less than 3.

Lines 8-11)   These lines contain parameters based on the size of the computational domain in line one to improve compiler performance of looping logic for vector computations. They must be present and not altered.

Line 13-14)   The logical MSMAGR controls the choice for the rotational suppression of turbulence within the sub-grid scale turbulence closure model.

True = Use the Richardson flux term with a rotational discriminator function.

False = Use the original Richardson flux term without a rotational discriminator function.

Line 16-17) The first three of these integer parameters control the activation of the massless tracer or DUST logic within TASS, and they must assume either the full domain size as shown on line 16 or assume the value of one as shown on the comment line 17. To disable DUST advection logic within TASS set these values to one. The next group of parameters used to control the size of the DUST array. The fourth (ND) parameter is still active and controls the number of dust domains. Each domain adds a prognostic equation to the simulation. The first domain (ND=1) is used for dust tracers or insects depending on DUSTON in the control file. If the simulation is of a multiple vortex interaction case, then ND-1 is the number of additional dust tracers that are used to identify and track individual vortex systems. To enable tracking of all individual vortex systems, ND should be set to VORTOT+1.

Line 19-20) These integer parameters control the activation of the precipitation logic within TASS, and they must assume either the full domain size as shown on line 19 or assume the value of one as shown on the comment line 20. To disable precipitation logic within TASS set these values to one.

Line 24-25) The logical parameter ORDER4 controls spatial accuracy for advection of momentum.

True = Use fourth-order spatial advection (recommended).

False = Use second-order spatial advection.

Line 29-30) The logical variable BELTRAMI enables Beltrami test simulations.

True = Run TASS in Beltrami mode.

False = No effect (recommended).

Line 34-35) The logical variable WAKE3D enables three-dimensional wake vortex simulations. When TASS assumes this mode, the upstream I plane (West boundary) assumes a fixed distribution corresponding to an initial wake system. This vortex system propagates into the domain as the domain translates toward the west with the movement of the generating aircraft. In this mode of simulation, each I station (along the *X*-coordinate) represents a different time for the vortex system corresponding to the domain translation speed. Note, UGRID in the control file should be set to the ground speed of the generating aircraft (with a negative value for the westward propagation).

True = Run TASS in three-dimensional wake vortex mode.

False = No effect (recommended).

Line 39-40) The logical parameter MIRROR sets a reflecting or mirror condition along the south boundary.

True = Reflect the domain.

False = No effect (recommended).

Line 44-45) The logical parameter LAMINAR controls how TASS handles the eddy viscosity of the simulation.

True = Run as DNS with a constant eddy viscosity value of VIS.

False = Run as LES (recommended).

Line 49-50) The logical parameter NOMOIST allows the user to disable all moisture processes.

True = No moist processes allowed.

False = No effect.

Line 54-55) The logical parameter FULLPER sets the vertical boundary condition.

True = All boundary conditions are fully periodic.

False = Lower boundary represents ground surface and top boundary is nonporous.

Line 59-60) The logical parameter NOSLIP sets the velocity condition at lower boundary when FULLPER is false.

True = A no-slip condition for velocity is imposed at the ground (recommended).

False = The lower boundary is a free-slip surface.

1)		INTEGER,PARAMETER :: NI=38, NJ=104, NK=102
2)		PARAMETER (IDIV = 1, JDIV = 8, KDIV = 4)
3)	C	MPI RELATED PARAMETERS
4)		INTEGER,PARAMETER :: MI = MIN (NI, NI/IDIV + 7)
5)		INTEGER,PARAMETER :: MJ = MIN (NJ, NJ/JDIV + 7)
6)		INTEGER,PARAMETER :: MK = MIN (NK, NK/KDIV + 7)
7)		INTEGER,PARAMETER :: NG=2048, NG1=NG-1,NGUTTER=3
8)		INTEGER,PARAMETER :: NK1=NK-1
9)		INTEGER,PARAMETER :: IS =NI ,JS =NJ ,KS =NK
10)		INTEGER,PARAMETER :: IS1=NI-1,JS1=NJ-1,KS1=NK-1
11)		INTEGER,PARAMETER :: IS2=NI-2,JS2=NJ-2,KS2=NK-2
12)	C	PARAMETER FOR SUBGRID SCALE TURB CLOSURE
13)		LOGICAL MSMAGR
14)		PARAMETER(MSMAGR=.TRUE.)
15)	C	PARAMETER FOR DUST
16)		PARAMETER (NID=NI, NJD=NJ, NKD=NK1, ND=1, NDD=ND)
17)	C	PARAMETER (NID=1 , NJD=1 , NKD=1 , ND=1, NDD=ND)
18)	C	PARAMETER FOR PRECIPITATION
19)		PARAMETER (NIP=NI, NJP=NJ, NKP=NK1)
20)	C	PARAMETER (NIP=1 , NJP=1 , NKP=1)
21)	C	
22)	C	FLAG FOR 4TH ORDER HORIZONTAL DERIVATIVES IN MOMENTUM
23)	C	
24)		LOGICAL ORDER4
25)		PARAMETER (ORDER4=.TRUE.)
26)	C	
27)	C	DATA FOR 3-D PERIODIC TEST CASE: ON/OFF
28)	C	
29)		LOGICAL BELTRAMI
30)		PARAMETER(BELTRAMI=.FALSE.)
31)	C	
32)	C	FLAG FOR THE SPECIAL BOUNDARY CONDITION FOR FULL 3-D WAKE VORTEX SYSTEM
33)	C	
34)		LOGICAL WAKE3D
35)		PARAMETER(WAKE3D=.FALSE.)
36)	C	
37)	C	FLAG FOR THE CROSSFLOW MIRROR BOUNDARY CONDITION FOR 3-D WAKE VORTEX SYSTEM
38)	C	
39)		LOGICAL MIRROR
40)		PARAMETER(MIRROR=.FALSE.)
41)	C	
42)	C	FLAG FOR LAMINAR (CONSTANT EDDY VISCOSITY) CALCULATIONS
43)	C	
44)		LOGICAL LAMINAR
45)		PARAMETER (LAMINAR=.FALSE., VIS=0.962 E-2)
46)	C	
47)	C	FLAG FOR TURNING OF ALL VAPOR AND MOIST PROCESSES
48)	C	
49)		LOGICAL NOMOIST
50)		PARAMETER (NOMOIST=.FALSE.)
51)	C	
52)	C	FLAG FOR THE FULL PERIODIC BOUNDARY CONDITION
53)	C	
54)		LOGICAL FULLPER
55)		PARAMETER(FULLPER=.FALSE.)
56)	C	
57)	C	FLAG FOR NOSLIP OR FREESLIP SURFACE BOUNDARY
58)	C	
59)		LOGICAL NOSLIP
60)		PARAMETER(NOSLIP=.TRUE.)

Figure 5-1. An example param.h include file.

The number of internal horizontal points in the domain is controlled by the boundary conditions for the lateral boundaries. For radiative or open boundaries, there are three boundary points (two on the East or North edges), and for periodic boundaries, there are four (two on all edges). The number of vertical boundary points is either three for a surface bounded domain (one below the surface) or five for a fully periodic domain (two below the periodic boundary).

5.1.2   Parameters in the include file area22.h

This file contains two parameters related to wake vortex simulations. The integer parameter NVORMAX sets the maximum number of vortex systems for a given simulation. This variable must be greater than or equal to the VORTOT variable within the control file. The other parameter is VFORCE. This parameter is valid only for three-dimensional wake vortex simulations and the variable INITYPE assumes the value of either 3 or 5. When active, it enables the forcing function option of the wake vortex simulation mode.

5.1.3   Parameters in the include file area17.h

This file contains the parameter THERMW. If this logical variable is true, then the temperature includes the contribution from the thermal wind. This only applies if all of the following logicals assume the value of false: UNHEAT, TSPEC, FLXSPEC, and EBUDG.

5.1.4   Parameters in the include file area10.h

This file contains the logical parameter FRONT, which sets the domain to represent a two-dimensional frontal boundary with periodic boundary conditions in the North-South direction. When true, this logical over-rides the value of the input logical SOPEN to false and prints an alert in the output file. The variables in the subroutine INITPER in the module *initial.f* that control the frontal boundary initialization are FDEPTH, FWIDTH, FTEMP, and FVPER.

5.1.5   Parameters in the include file fftw_f77.h

The file fftw_f77.h contains parameter values that directly affect the FFTW fast Fourier transform behavior, and the user should consult freely available documentation for those variables located at www.fftw.org. The user should not alter the values unless thoroughly familiar with the effects of these variables.

5.1.6 <u>Parameters in the include file heat.h</u>

The first parameter in this file is ITDIM. This sets the maximum number of data items for planetary boundary layer (PBL) heating time history specification that may be read in. The code prints an error message and terminates the execution if the user requests a number larger than ITMAX in the PBL input file. There is no printout if ITMAX assumes an acceptable value. The other parameter in this file is the integer IGDIM, which sets the limit of the number of geostrophic wind profiles (NGEOTIM). This value only appears in the output if exceeded, which terminates the simulation.

5.2 Parameters within the TASS Code

The remaining parameters are contained within the TASS code modules. Table 5-1 contains the important parameters contained within the TASS code along with an explanation of each one.

Table 5-1. TASS parameters imbedded in the code

Module	Parameter	Description
driver.f	ABC	Modified Adams-Bashforth time integration constant for the large time step.
	EXTFORCE	A logical variable that turns on/off the external forcing of the rate terms of the velocity components using the FFT routines in z*efft.f.* (used for generating von-Karman turbulence).
	GRAVWAV	When true, average buoyancy term between N and N+1 time step; otherwise, uses buoyancy at N+1 time step (recommended).
	ISTPCHK	This integer sets the minimum number of iterations before calling <u>FDIV</u> for diagnostic checks. It primarily allows TASS to get beyond transient dynamics.
	MXITR	Maximum number of time steps allowed.
	NCHECK	Every NCHECK iterations the time step stability criteria is checked, and diagnostic information is appended to the main output file.
	NQTIMES	The maximum number of individual timing categories to monitor. TASS currently monitors 36 different timing categories and sub-categories in the subroutine.
	REGRID	This logical controls the writing of a restart file if TEST is true. If true, TASS writes a restart file; otherwise it does not.
	TIMEZERO	When TASS re-initializes a simulation, this logical controls the continuity of the time for the simulation. If true, TASS resets the time to zero after re-initialization, otherwise time continues from previous value acquired from the restart file.
	ITORLBC	An integer flag to control the boundary condition for potential temperature. Defaults to 1, which signifies Orlanski radiative boundary conditions. If set to 0, advective outflow boundary conditions apply.

Module	Parameter	Description
advectl.f	THRS THRSI THRSIN THRSN	These are threshold values used for water substance used in "hole filling" when not using the TVD advection option.
	TVD_ON	A logical to enable use of the TVD advection scheme.
	FLXMN	A real variable that sets the minimum value of flux used by the surface energy budget routines.
bounds.f	FLXMX	Same as FLXMN, but for the maximum value.
	QH	Coefficient to control the magnitude of precipitation falling through the top boundary. This is a local variable defined in subroutine BNDRYZ and varies from 0 to 1.
	TRAT	A real variable that sets the minimum value for precipitation values to insure division operations do not produce a floating point exception.
	TTRS	A real variable to set the maximum value of the ratio of the precipitation substances before and after subroutine <u>HOLE</u> resets their values.
	THLD THLDN	Thresholds used by the Orlanski boundary routines. These are set to extremely small numbers, and the user must insure that they are representable by the FORTRAN compiler.
	THRSR THRSS	Threshold constants used in "hole filling".
	DEWOUT	A logical variable to control the data type output to the graph file for the water vapor variable (XIV). If true, outputs dewpoint temperature (°C), and if false outputs XIV.
dataproc.f	AMBVORT	This logical controls how subroutine <u>PLOTV</u> computes the vorticity. When true, PLOTV removes the vorticity from the initial profile of the U component of velocity.
	PERROR	The percent difference between the area and the actual calculated area at a given radius. This is used in PLOTV to determine when a circulation computation is invalid.
	SHORTOUT	This logical variable controls the amount of information written to the vortex tracking output files (thvn and thvp files, refer to section 4 for details of the contents of these files). When false, <u>PLOTV</u> prints all available circulation related data.
	SKIP_****	This logical variable designation refers to numerous logical variables used to control output to the graph file. The variables assume the form shown, where "****" takes the name of the specific output variable option. The user shall edit these values to obtain the desired output variables in the graph file.
	ZHGT	This real variable sets the height the subroutine <u>TKEHIST</u> monitors turbulent kinetic energy for non-fully periodic simulations when EXTFORCE is true.
	HZRIC	If true, the contribution from the horizontal terms is added to the vertical terms in the Richardson number.

Module	Parameter	Description
diagvar.f	SRICH	If true, apply smoothing to the Richardson number.
	MSMAGR	A logical to control the use of the modified Smagorinski model which adds terms to account for the rotational effects upon eddy viscosity in the Richardson flux term.
	CROT	A constant used on the rotational component of the Smagorinski model. Used only if MSMAGR is true.
	RCRTIN	Threshold for the lower limit of the Richardson flux number (Rf).
	RFCRIT	Threshold for the upper limit of the Richardson flux number (Rf) based on the relation: 1-Rf 65 RFCRIT. RFCRITR applies when rain or hail is present.
	VTHRS	This real variable sets the maximum groundspeed before TASS aborts a simulation.
	VREMAN	When true, this logical variable enables use of the Vreman sub-grid scale turbulence closure model (Vreman, 2004).
	SPLF	A value used to control the tension of the spline interpolation of the sounding. SPLF = 0,1, and >50 correspond to a cubic spline, a cubic spline under tension, and approximately linear interpolation, respectively. Values less than 10 are not recommended.
initial.f	IENDSW	A parameter controlling the end conditions of the interpolated spline. See subroutine STIUNI in module *util.f* for further explanation.
	ENVSAT	A logical variable to allow a saturated environment. If true, maximum relative humidity of the environment is allowed up to 100%; otherwise maximum humidity is allowed only up to 95%.
	NDSNDG	This logical variable controls how the subroutine <u>INITIAL</u> interprets the value of NBRUNT. When true, NBRUNT assumes the non-dimensional value generated from wake vortex data in the input file regardless of the value of HVORT. Otherwise, NBRUNT assumes the dimensional value.
	NEWSND	A logical variable used to generate a new sounding based on the surface temperature and the value of the Brunt-Vaisala frequency, NBRUNT. The sounding is printed out to FORTRAN unit 23 and the code then exits.
	NBRUNT	The Brunt-Vaisala frequency. Used only if NEWSND is true.
	EDBACK	Background eddy viscosity subject to the following relation: $$EDBACK= \begin{cases} 7.5\text{x}10^{-5}\Delta^2 & \Delta_m \geq 200m \\ 3.5\text{x}10^{-5}\Delta^2 & 200m > \Delta_m \geq 40m \\ 2.5\text{x}10^{-6}\Delta^2 & 40 > \Delta_m \end{cases}$$ where $\Delta_m = \min(\Delta x, \Delta y)$ and Δ=horizontal grid size defined in subroutine DATA.
	FTWOGRD	When true, assumes sub-grid length scale is proportional to the minimum resolvable scale, 2Δ. If false, length scale is proportional to the grid size, Δ.

Module	Parameter	Description
	PHGT	This logical variable controls the contents of the newZZ sounding file when NEWSND is true. If true, the sounding heights are in pressure; otherwise they are the elevation above the surface.
	PR	Turbulent Prandtl number.
	RHOCNST	When true, this logic variable sets the density to a constant value.
	SPR	Turbulent Prandtl number at the surface.
	SPRDCNS	This real variable sets the constant for vertical spreading of drops and hail particles resulting from size distorting (used in <u>MARCHL</u>)
	TSTEADY	Allows initial temperature profile to be balanced by residual terms. Used only if NOSTEADY is true.
	SEC	Sub-grid turbulence closure coefficient.
	XN	Length matching scale at the ground (Proctor 1996).
	FIL6C	Sixth-order spatial filtering coefficient. For details see Switzer, 1996. If set to zero, no sixth-order filtering is performed.
	FILCNS	Constant used for top and lateral boundary filtering. Set to zero to disable boundary filtering.
	IWAVE JWAVE KWAVE	These integers set the number of wave initialized for a Beltrami test simulation.
	WMAG	This real variable sets the magnitude of the vertical velocity for a Beltrami simulation.
	GDNY	Density of Graupel ($kg\ m^{-3}$).
	SDNY	Density of Snow ($kg\ m^{-3}$).
	V75	The aircraft velocity used in the computation of the F-factor. Currently set at 75 m/s.
	UPOLY	A logical to control the setting of the U component of the velocity profile. If true the velocity is set to a polynomial. See the subroutine INITIAL for further explanation.
	AXIAL	A logical to enable the initialization of axial flow in three- dimensional wake vortex systems. If true the axial flow set based on logic in the code otherwise the velocity in the axial direction is set to zero.
initper.f	FWIDTH FDEPTH FTEMP FVPER	These four real variables control the characteristics of the two-dimensional cool pool/front when FRONT is true.
	IFORCON IFORPOS IFORSTRT	These variables are applicable only for three-dimensional wake vortex simulations using the forcing function option. IFORCON is an array of strength coefficients used upon the two-dimensional vortex field as the forcing function. IFORPOS is the I-plane about which the forcing function is to be centered, and IFORSTRT is the number of I-planes in front of IFORPOS where IFORCON is to be applied from. The forcing function option is not recommended.
	ISEED	Random number generator seed value for turbulence generation.

Module	Parameter	Description
	AIAA AIAAM LAMB RANKIN HJOUB	These are logical variables to control the type of two-dimensional vortex system with which to initialize. Only one may be true. The first two are the name of the scheme. The AIAA is the Burnham- Hallock method and the AIAAM is similar to the AIAA model, but is based upon observations of several wake vortices measured early in their evolution. The details of this model is contained in Proctor (1998). The third option is the Lamb vortex. The fourth option is the combined Rankin vortex. The remaining option, HJOUB, is the Hoffman-Jourbet method and is not recommended. The AIAAM model is the recommended option to use for vortex initialization.
	EXTENT	This real variable sets the maximum radius of influence used for wake vortex initialization. (set 1×10^{11} m)
	IMGLEVELS	This integer variable sets the number of wake vortex image levels for initialization.
	NOCORNER	A logical which if true initializes the domain with the effect of image vortex systems that are either directly above or laterally to the side. All other image systems in the image domain are ignored.
	NOIMGVOR	A logical which if true initializes the domain without any image vortex systems.
	PEPS NSTOP	These two variables control the Successive over-relaxation iterations in subroutine PINIT. The real variable PEPS sets the convergence criteria for a solution, and the integer variable NSTOP sets the maximum number of iterations for the process.
	SINGVORT	A logical which if true initializes the domain with only an isolated vortex.
	SLABS	If true and NJ is less than 12, the initial thermal is set to be two-dimensional. Otherwise the initial thermal is three-dimensional.
	SHAFT	If true precipitation is initialized as a shaft with the center at XCENTER and YCENTER. If false the precipitation is set to be uniform.
	VELPERT DELTAR	VELPERT is a logical which if true adds a random perturbation to the velocity field. The amplitude of the perturbation is set by DELTAR.
	TRPRESS	A variable to control how multiple vortex systems are traced. If less than 0, tracing is disabled. 1=trace pressure, 0= trace vorticity. The option for multiple vortex systems is still under testing.
	NOWRITE	If true, this logical variable disables writing to the graph file.
io.f	CTRACK	This real variable, which assumes the value of either 1 or –1 controls the tracking of a storm. If set to 1. TASS tracks the cyclonic rotating updraft; otherwise TASS tracks the anti-cyclonic rotating updraft.
march.f	EPSTM EPSTMS	Coefficient for Rayleigh filter on top boundary for large and small time steps, respectively. EPSTM applies to potential temperature and EPSTMS applies to vertical velocity. To disable this function, set to zero.

Module	Parameter	Description
	KEEPSND	A logical, which if true, causes the mean profile to be nudged toward the base-state profiles for U, V, W, and THETA. Applied at every large time step to maintain mean profile for sounding when turbulence is present.
	UTHLD VTHLD TTHLD	Real variables to control the magnitude of nudging upon the environmental sounding when KEEPSND is true.
	RAYGRV	Coefficient for Rayleigh filter for normal velocity components just inside of lateral boundaries. To disable this function, set to zero.
	RAYP	Relaxation factor used in radiative boundary conditions. Refer to page 24 and 25 in Proctor 1987a for details. Set to zero to disable.
	RAYBC RECTM RECTMS	Relaxation factor used in radiative boundary conditions for large and small time steps, respectively. RAYBC and RECTM apply to normal velocity components on lateral boundaries and RECTMS applies to pressure. Refer to page 24 and 25 in Proctor 1987a for details. Set to zero to disable.
	STBCNS1 STBCNS2	Constants to insure linear time marching stability for velocity and scalar variables, respectively.
	CAMP	This real variable, used for turbulence simulations, sets the magnitude of the energy added to the rate terms. This value corresponds to the eddy dissipation rate for the turbulence domain.
zefft.f	CWAVEN	This real variable, used for turbulence simulations, sets the minimum wave number for which TASS adds energy to the rate terms.

5.3   TASS Debugging

The above parameters are a grouped list based on location within the code. However, they can also be grouped according to the function that they control. This section presents some groupings that can be used to disable specific components of TASS.

5.3.1   <u>Filtering</u>

There are several different filters used in TASS, ranging from boundary filters to domain filters. The boundary filters are Rayleigh filters and sponges. The Rayleigh filters are controlled by the variables EPSTM, EPSTMS, and RAYGRV; the boundary sponges are controlled by the variables RECTM, RECTMS, RAYP, and FILCNS. To disable the boundary filtering and sponging, these variables all need to be set to zero. The domain filter is controlled by the variable FIL6C.

Setting this variable to zero disables domain filtering. Note that a choice of periodic boundary conditions disables lateral boundary filters.

5.3.2   Sub-grid turbulence closure

Direct Numerical Simulations (DNS) are allowed by setting the logical variable LAMINAR to true in the param.h file and setting the value of VIS in param.h to the constant viscosity. If VIS is chosen smaller than the value for molecular viscosity, the later will be used. Otherwise, Large Eddy simulations with sub-grid closure are performed.

5.3.3   Microphysics

There are different levels of functionality for this option. To disable all microphysics, set the logical variables DRYNEU and MICRON to true and false, respectively. This automatically forces PRECIPON to be false and also modifies the input sounding to be dry and neutral. If the user wants to disable all microphysics and moisture processes without modifying the sounding, set the parameter NOMOIST to be true. Further details regarding parametric setting are contained in Table 5-2.

5.3.4   Coriolis force

Setting both DLAT and AMVORT to zero disables the Coriolis force; otherwise, the Coriolis force is computed from DLAT in the control file.

Table 5-2. Microphysics parameter settings

	ICEON	MICRON	PRECIPON	NOMOIST	DRYNEU
Full microphysics	T	T	T	F	F
No ice processes	F	T	T/F	F	T
No precipitation processes	T/F	T	F	F	F
No moisture processes with water vapor	not used	F	F	F	F
No moisture processes - dry processes only	not used	not used	not used	T	F
Dry neutral atmosphere with no moist processes	not used	not used	not used	not used	T

6.0 TASS OPERATIONAL CONFIGURATIONS

This section gives an overview of the different ways TASS can be configured.

6.1 Atmospheric Simulation

This mode is useful for simulation of the following situations: convective local storms, microburst/windshear, hailstorms, and tornadic thunderstorms. If a modification to the preset time ramping of the precipitation falling through the top boundary is desired, the user will have to edit subroutine <u>BNDRYZ</u> in module *bounds.f*.

6.2 Planetary Boundary Layer

This mode enables simulation of the planetary boundary layer by the specification of time dependent or constant surface heat flux information.

6.3 Aircraft Wake Vortex

In this configuration, TASS starts with a rolled up vortex pair and simulates the subsequent motion and environmental interaction. The vortex system may either be two-dimensional or three-dimensional. The three-dimensional system is still under development. Multiple vortex systems may be simulated for two-dimensional simulations only.

6.4 Two-Dimensional or Slab

The above three modes may be run as two-dimensional. This is achieved by the setting of the parameter NJ in the param.h include file to 5.

6.5 Beltrami

This mode simulates the three-dimensional Beltrami flow field and outputs information about the accuracy of the TASS model. Details are in Switzer, 1996.

7.0 TASS UTILIZATION

7.1 Program Structure

TASS is a FORTRAN program comprised of two sets of files. The first set is the modules that make up the body of the code. These files have been explained in section 2.4. The other group consists of the include files. The compiler incorporates these include files within TASS at compilation time. The purpose of include files is to contain in one location the definition of the FORTRAN common blocks and parameters. Therefore, when a change is made in an include file, the change is automatically propagated throughout the TASS code upon compilation. These files are, by default, located in a subdirectory, "Include", of the TASS source directory. Editing the path for each "include" line in each of the modules changes this default location.

7.2 How to Compile and Run TASS

TASS is compiled using a makefile on unix systems. The makefile is executed by typing "make" at the system prompt. The makefile contains dependency information for the run-time files (object and executable) based on the source files (include and module). Therefore, the advantage of using a makefile is selective compilation based on the need to update only those parts of the code that have changed. The makefile operates by first determining which source files have changed— their modification date is older than run-time files. The makefile then selectively compiles the necessary source files to bring the run-time files up to date.

To run the executable, the control file and the sounding file must be present in the local directory. In addition, optional files for planetary boundary layer simulations or restart runs may also need to be present. The main output file will be printed to the terminal. Therefore, to save this file, redirect the terminal output to a file. Generally FORTRAN units 5 and 6 are standard input and output, respectively. A sample command to run TASS with this input output convention is:

tass < control.file > output.file

After the run, all of the files described in section 4.0 will be present in the directory where TASS was run.

To introduce an orderly control of all the TASS output files a C shell script exists that creates the necessary links to these files so that when the simulation terminates, they all have predefined

names different from the original default names. A copy of this file should be present in the source code directory as the file named "run_tass".

7.3 TASS Run Checklist

The following list presents one possible order in which to put together a TASS simulation run. This list also serves as an outline whereby the user may refer to the appropriate section to learn further about specific variables and requirements.

1) Decide on the type of simulation to be performed.

2) Determine the boundary conditions and then the desired domain size (see section 5.1).

3) Determine the number of processors to use and the shape of the processor grid.

4) Set the parameters in the include file param.h (see section 5.1).

5) Check and set the appropriate parameters within the code (see section 5.2).

6) Create a sounding to initialize the domain and insure that the file is readable by FORTRAN unit 9 (see section 3.2).

7) Edit the control file to obtain the appropriate simulation run (see section 3.1).

8) If the simulation is a planetary boundary layer simulation, create the corresponding input file (see section 3.3).

9) If the run is to be restarted from a previous TASS run, insure that the restart file is readable by FORTRAN unit 3 (see section 3.4).

10) Edit the module dataproc.f and examine the subroutine GRAPH to insure that the desired variables are going to be output.

11) Compile the code using the makefile.

12) Set the input logical variable TEST to true and run the code (see section 3.1).

13) Check the main output file and, if possible, the graph files to verify the initial domain is as expected (see sections 4.1 and 4.3).

14) Reset TEST to false, and re-run the simulation.

15) If the simulation is to be restarted, then move the fort.2 or output restart file to fort.3 or the input restart file (see sections 3.4 and 4.2).

16) Re-run TASS from the restart file as necessary to complete the entire simulation.

REFERENCES

Ahmad, N., and F.H. Proctor, 2011: Advection of microphysical scalars in Terminal Area Simulation System (TASS). AIAA 2011-1004.

Bowles, R. L., 1990: Reducing windshear risk through airborne systems technology. The 17th Congress of the ICAS, Stockholm, Sweden, 27 pp.

Cline, A. K., 1974: Scalar- and planar-valued curve fitting using splines under tension. *ACM Comm.*, Vol. 17, No 4, pp. 218-223.

Carpenter, K. M., 1982: Note on the paper "Radiation conditions for the lateral boundaries of limited-area numerical models." *Quart. J. R. Met. Soc.*, Vol. 108, 717-719 pp.

Clark, T. L. and R. D. Farley, 1984: Severe downslope windstorm calculations in two and three spatial dimensions using anelastic interactive grid nesting: A possible mechanism for gustiness. *J. Atmos. Sci.*, Vol. 41, No 3, pp. 329-350.

Fitzgerald, J. W., and P. A. Spyers-Duran, 1973: Changes in cloud nucleus concentration and cloud droplet size distribution associated with pollution from St. Louis. *J. Appl. Meteorol.*, Vol. 12, pp. 511-516.

Haltiner, G. J. and Williams, R. T., 1980: *Numerical Prediction and Dynamic Meteorology*, second edition. John Wiley & Sons, pp. 226-230.

Klemp, M. L., and R. B. Wilhelmson, 1978: The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, Vol. 35, pp. 1070-1096 .

Knight, C. A., W. A. Cooper, D. W. Breed, I. R. Paluch, P. L. Smith, and G. Vali, 1982: Microphysics, hailstorms of the Central High Plains, Vol. 1. National Hail Research Experiment. C. A. Knight and P. Squires, Eds., Colorado Associated University Press, Boulder, pp. 151-193.

Leonard, B. P., 1979: A Stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comp. Meth. Appl. Mech. Eng.*, Vol. 19, pp. 59-98.

Press, W. H., W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery, 1992: *Numerical recipes in FORTRAN, the art of scientific computing*, second edition. Cambridge University Press, p. 271.

Proctor, F. H., 1987: The Terminal Area Simulation System, Volume I: Theoretical formulation. NASA CR-4046, 176 pp.

Proctor, F. H., 1996: Numerical simulation of wake vortices measured during the Idaho Falls and Memphis field programs. 14th AIAA Applied Aerodynamics Conference, Proceedings, Part-II, AIAA. 96-2496, pp. 943-960.

Proctor, F. H., 1998: The NASA-Langley wake vortex modelling effort in support of an operational aircraft spacing system. AIAA 98-0589.

Proctor, F.H., D.A. Hinton, and R.L. Bowles, 2000: A windshear hazard index. 9th Conference on Aviation, Range and Aerospace Meteorology, Amer. Meteor. Soc., 482-487 pp.

Proctor, F.H., and D.W. Hamilton, 2009: Evaluation of fast-time wake vortex prediction models. AIAA 2009-0344.

Schowalter, D. G., D. S. DeCroix, Y.-L. Lin, S. P. Arya, and M. Kaplan, 1996: Planetary boundary layer simulation using TASS. NASA CR-198325, 41 pp.

Simpson J., and V. Wiggert, 1969: Models of precipitating cumulus towers. *Mon. Wea. Rev.*, Vol. 97, pp. 471-489.

Switzer, G. F., 1996: Validation tests of TASS for application to 3-D vortex simulations. NASA CR-4756, 45 pp.

Van Leer, B., 1979: Towards the ultimate conservative difference scheme: V, A second-order sequel to Godunov's Method. *Journal of Computational Physics*, Vol. 32, pp. 101-136.

Vreman, A.W., 2004: An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids*, Vol. 16, pp. 3670-3681.

APPENDIX

BIBLIOGRAPHY OF STUDIES THAT HAVE UTILIZED TASS

CUMULUS CONVECTION AND SEVERE LOCAL STORMS

Deal, R. L., 1996: A Numerical investigation into the dynamics of the Palm Sunday 1994 supercell thunderstorms. Master's Thesis, North Carolina State University, 105 pp.

Deal, R. L., Y.-L. Lin, M. S. Kulie, and D. S. DeCroix, 1996: Observations and simulation of a long-lived tornadic storm: The 27 March 1994 Cherokee county, Alabama supercell. Preprints, 18th Conference on Severe Local Storms, San Francisco, CA, Amer. Meteor. Soc., Feb.

Elmore, K.L., 2000: Ensemble cloud model applications to forecasting thunderstorms. Ph.D. Dissertation, University of Oklahoma, Oklahoma City, OK, 148 pp.

Elmore, K.L., P.L. MacKeen, L. McKibben, F.H. Proctor, and H.E. Brooks, 1997: Thunderstorm growth and decay forecasting using 2-D ensemble modelling. 7th Conference. on Aviation, Range, and Aerospace Meteorology, Amer. Meteor. Soc.

Kulie, M. S., 1996: The structure and evolution of a numerically simulated high-precipitation supercell thunderstorm. Master's Thesis, North Carolina State University, 119 pp.

Kulie, M. S., Y.-L. Lin, R. L. Deal, and D. S. DeCroix, 1996: A cloud-scale numerical simulation of the 28 November 1988 Raleigh tornadic thunderstorm. Preprints, 18th Conference on Severe Local Storms, Amer. Meteor. Soc., pp. 283-287.

Proctor, F. H., 1983: Numerical simulation of a bell-shaped cumulonimbus. Preprints, 13th Conf. on Severe Local Storms, Amer. Meteor. Soc., pp. 235-240.

Proctor, F. H., 1984: Small-scale CISK as a mechanism for isolated cumulonimbus development. International Conf. on Mesoscale Meteorology, Melbourne, Australia, 13 pp.

Proctor, F. H., 1986: Three-dimensional simulation of the 2 August CCOPE hailstorm with the Terminal Area Simulation System. Report of the International Cloud Modelling Workshop/Conference, Irsee FRG, WMO Tech. Document No. 139, pp. 227-240.

Proctor, F. H., 1987: The Terminal Area Simulation System. Volume I: Theoretical formulation. NASA CR-4046, 176 pp.

Proctor, F. H., 1987: The Terminal Area Simulation System. Volume II: Verification experiments. NASA CR-4047, 112 pp.

Proctor, F. H., 1988: Numerical simulations of an isolated microburst. Part I: Dynamics and structure. *J. Atmos. Sci.*, Vol. 45, pp. 3137-3160.

Proctor, F. H., 1989: Numerical simulations of an isolated microburst. Part II: Sensitivity experiments. *J. Atmos. Sci.*, Vol. 46, pp. 2143-2165.

Proctor, F., N. Ahmad and F. Limon-Durparcmeur, 2012: Numerical Simulation of a Tornado Generating Supercell. AIAA 2012-0557.

WINDSHEAR/MICROBURST

Arbuckle, P. D., M. S. Lewis, and D. A. Hinton, 1996: Airborne systems technology application to the windshear threat. 20th Congress International Council of the Aeronautical Sciences, Paper No. 96-5.7.1, 11 pp.

Bracalente, E. M., C. L. Britt, and W. R. Jones, 1990: Airborne Doppler radar detection of low-altitude wind shear. *Journal of Aircraft*, Vol. 27, No 2, pp. 151-157.

Byrd, G. P., F. H. Proctor, and R. L. Bowles, 1990: Evaluation of a technique to quantify microburst windshear hazard potential to aircraft based on data from a windshear model. 29th IEEE Conf. on Decision and Control, IEEE Control Systems Society, pp. 689-695.

Chuang, S., F. H. Proctor, J. W. Zack, and M. L. Kaplan, 1984: A numerical weather prediction system designed to simulate atmospheric downburst phenomena. AIAA 84-0352.

Proctor, F. H., 1985: Numerical simulation of precipitation induced downbursts. Preprints, Second International Conf. on the Aviation Weather System, Amer. Meteor. Soc., pp. 257-264.

Proctor, F. H., 1988: Numerical simulations of an isolated microburst. Part I: Dynamics and structure. *J. Atmos. Sci.,* Vol. 45, pp. 3137-3160.

Proctor, F. H., 1988: Numerical simulation of the 2 August 1985 DFW microburst with the three-dimensional Terminal Area Simulation System. Preprints Joint Session of 15th Conf. on Severe Local Storms and Eighth Conf. on Numerical Weather Prediction, Amer. Meteor. Soc., J99-J102 pp.

Proctor, F. H., 1989: A relation between peak temperature drop and velocity differential in a microburst. Preprints Third International Conf. on the Aviation Weather System, Amer. Meteor. Soc., pp. 5-8.

Proctor, F. H., 1989: Numerical simulations of an isolated microburst. Part II: Sensitivity experiments. *J. Atmos. Sci*., Vol. 46, pp. 2143-2165.

Proctor, F. H., 1993: Case study of a low-reflectivity pulsating microburst: numerical simulation of the Denver, 8 July 1989, storm. Preprints, 17th Conf. on Severe Local Storms, Amer. Meteor. Soc., pp. 677-680.

Proctor, F. H., 1994: Investigation of microburst windshear associated with the Charlotte 1994 accident using a meteorological numerical model. National Transportation Safety Board, Docket No. SA-509, Exhibit No. 5-I, 31 pp.

Proctor, F. H., and R. L. Bowles, 1989: Investigation of the Denver 11 July 1988 microburst storm with the three-dimensional NASA-Langley Windshear Model. Appendix 2 of Windshear Case Study: Denver, Colorado, July 11, 1988 -- Final Report, DOT/ FAA/DS-89/19.

Proctor, F. H., and R. L. Bowles, 1990: Three-dimensional simulation of the Denver 11 July storm of 1988: An intense microburst event. Preprints, 16th Conference on Severe Local Storms, Amer. Meteor. Soc., 373-378 pp.

Proctor, F. H., and R. L. Bowles, 1992: Three-dimensional simulation of the Denver 11 July 1988 microburst-producing storm. *Meteorol. and Atmos. Phys*., Vol. 49, pp. 107-124.

Proctor, F. H., E. M. Bracalente, S. D. Harrah, G. F. Switzer, and C. L. Britt, 1995: Simulation of the 1994 Charlotte microburst with look-ahead windshear radar. Preprints, 27th Conference on Radar Meteorology, Amer. Meteor. Soc., pp. 530-532.

Proctor, F.H., D.A. Hinton, and R.L. Bowles, 2000: A windshear hazard index. 9th Conference on Aviation, Range and Aerospace Meteorology, Amer. Meteor. Soc., 482-487 pp.

Switzer, G. F., F. H. Proctor, D. A. Hinton and J. V. Aanstoos, 1993: Windshear database for forward-looking systems certification. NASA TM-109012, 133 pp.

Vicroy, D. D., 1991: A simple, analytical, axisymmetric microburst model for downdraft estimation. NASA TM-104053, DOT/FAA/RD-91/10.

Vicroy, D. D., 1992: Assessment of microburst models for downdraft estimation. *Journal Aircraft*, Vol. 29, No. 6, pp. 1043-1048.

Vicroy, D. D., 1994: Microburst vertical wind estimation from horizontal wind measurements. NASA TP-3460, DOT/FAA/RD-94/7.

PLANETARY BOUNDARY LAYERS

DeCroix, D. S., Y.-L. Lin, and D. G. Schowalter, 1997: Cellular convection embedded in the convective planetary boundary layer surface layer. *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 67, pp. 387-401.

DeCroix, D.S., 2001: Large-Eddy simulations of the convective and evening transition planetary boundary layers. Ph.D. Dissertation, North Carolina State University, 275 pp.

DeCroix, D.S., D.G. Schowalter, Y.-L. Lin, S.P. Arya, and F.H. Proctor, 1997: A three-dimensional nested grid large-eddy simulation of the convective planetary boundary layer. Preprints, 12th Symposium on Boundary Layers and Turbulence, pp. 210-211.

Ding, F., 2000: A subgrid-scale model for the large-eddy simulation of atmospheric boundary layer. Ph.D. dissertation, North Carolina State University.

Ding, F., S.P. Arya, and Y.-L. Lin, 2001: Large-eddy simulations of the atmospheric boundary layer using a new subgrid-scale model, I: slightly unstable and neutral cases. Environmental Fluid Mechanics, Vol. 1, pp. 29-47.

Ding, F., S.P. Arya, and Y.-L. Lin, 2001: Large-eddy simulations of the atmospheric boundary layer using a new subgrid-scale model, II: weakly and moderately stable cases. Environmental Fluid Mechanics, Vol. 1, pp. 49-69.

Schowalter, D. G., D. S. DeCroix, Y.-L. Lin, S. P. Arya, and M. L. Kaplan, 1996: The sensitivity of large-eddy simulation to local and nonlocal drag coefficients at the lower boundary. NASA CR-198310, 43 pp.

Schowalter, D. G., D. S. DeCroix, Y.-L. Lin, S. P. Arya, and M. Kaplan, 1996: Planetary boundary layer simulation using TASS. NASA CR-198325, 41 pp.

Schowalter, D. G., D. S. DeCroix, Y.-L. Lin, F. H. Proctor, S. P. Arya, and M. L. Kaplan, 1995: Turbulent statistics in the atmospheric boundary layer: a comparison of large-eddy simulation with observations. Preprints, 11th Symposium on Boundary Layers and Turbulence, Amer. Meteor. Soc., pp. 552-555.

Schowalter, D. G., Y.-L. Lin, and S. P. Arya, 1996: The evening transition of the planetary boundary layer: A case study. Procedings First Internationl Conference Nonlinear Problems Aviation and Aerospace.

CONVECTIVE INDUCED TURBULENCE

Ahmad, N., and F. Proctor, 2011: Large Eddy Simulations of Severe Convective Induced Turbulence. AIAA 2011-3201.

Hamilton, D.W., and F.H. Proctor, 2002: Convectively-Induced Turbulence Encountered During NASA's Fall-2000 Flight Experiments. 10th Conference on Aviation, Range, and Aerospace Meteorology, Amer. Meteor. Soc., pp. 371-374.

Hamilton, D.W. and F.H. Proctor, 2003: An aircraft encounter with turbulence in the vicinity of thunderstorm. AIAA 2003-4075.

Hamilton, D.W., and F.H. Proctor, 2006: Airborne Turbulence Detection System Certification Tool Set. AIAA 2006-0075.

Hamilton, D.W., and F.H. Proctor, 2006: Progress in the Development of an Airborne Turbulence Detection System, 12th Aviation, Range, and Aerospace Meteorology Conference, Amer. Meteor. Soc., 9 pp.

Hamilton, D.W., F.H. Proctor, and N.N. Ahmad, 2012: Flight Tests of the Turbulence Prediction and Warning System (TPAWS). NASA TM-2012-217337, February 2012, 152 pp.

Proctor, F.H., D.W. Hamilton, and R.L. Bowles, 2002: Numerical Study of a Convective Turbulence Encounter. AIAA Paper No. 2002-0944.

Proctor, F.H., D.W. Hamilton and R.L. Bowles, 2002: Numerical Simulation of a Convective Turbulence Encounter. Preprints 10th Conference on Aviation, Range, and Aerospace Meteorology, Amer. Meteor. Soc., pp. 41-44.

AIRCRAFT WAKE VORTICES

Han, J., 1998: Large eddy simulations of aircraft wake vortices in a homogeneous atmospheric turbulence. Ph.D. dissertaion, North Carolina State University, 171 pp.

Han, J., Y.-L. Lin, S. P. Arya , and F. H. Proctor, 1999: Large eddy simulation of aircraft wake vortices in a homogeneous atmospheric turbulence: Vortex decay and descent. AIAA 99-0756.

Han, J., Y-L. Lin, S.P. Arya, and F.H. Proctor, 2000: Numerical Study of Wake Vortex Decay and Descent in a Homogeneous Atmospheric Turbulence, AIAA Journal, Vol 38., No. 4, 2000, pp. 643-656.

Han, J., Y-L, Lin, S.P. Arya, S. Shen, and F.H. Proctor, 1999: Decay of Aircraft Wake Vortices in a Homogeneous Atmospheric Turbulence: A Large Eddy Simulation Study, 13th Conf. on Boundary Layers and Turbulence, Amer. Meteor. Soc.

Han, J., Y.L. Lin, D.G. Schowalter, S.P. Arya, and F.H. Proctor, 1997: Large-eddy simulation of aircraft wake vortices: Atmospheric Turbulence Effects. 12th Symposium on Boundary Layers and Turbulence, Univ. of British Columbia, Vancouver BC, Canada, July 28 - August 2, Amer. Meteor. Soc., pp. 237-238.

Han, J., Y-L. Lin, D.G. Schowalter, S.P. Arya, and F.H. Proctor, 2000: Large Eddy Simulation of Aircraft Wake Vortices Within Homogeneous Turbulence: Crow Instability. AIAA Journal. Vol. 38, No. 2, February, pp. 292-300.

Lin, Y-L, J. Han, J. Zhang, F. Ding, S.P. Arya, and F.H. Proctor, 2000: Large Eddy Simulation of Wake Vortices in the Convective Boundary Layer. AIAA Paper No. 2000-0753.

Marshall, R. E., W. Scales, and T. Myers, 1996: Wake vortex radar reflectivity. Research Triangle Institute Technical Memorandum, 45 pp.

Marshall, R. E., and T. J. Myers, 1996: Wingtip generated wake vortices as radar targets. IEEE 1996 National Radar Conference Proceedings, pp. 184-189.

Proctor, F. H., 1996: Numerical simulation of wake vortices measured during the Idaho Falls and Memphis field programs. AIAA 96-2496.

Proctor, F. H., 1998: The NASA-Langley wake vortex modelling effort in support of an operational aircraft spacing system. AIAA 98-0589.

Proctor, F.H., 2009: Interaction of Aircraft Wakes from Laterally Spaced Aircraft. AIAA 2009-0343.

Proctor, F., N. Ahmad, 2011: Crosswind Shear Gradient effect on Wake Vortices. AIAA 2011-3038.

Proctor, F.H., N.N. Ahmad, G.S. Switzer, and F. M. Limon Duparcmeur, 2010: Three-Phased Wake Vortex Decay.   AIAA 2010-7991.

Proctor, F. H. and J. Han, 1999: Numerical study of wake vortex interaction with the ground using the Terminal Area Simulation System. AIAA 99-0754.

Proctor, F.H., D.W. Hamilton, and J. Han, 2000: Wake Vortex Transport and Decay in Ground Effect: Vortex Linking with the Ground, AIAA 2000-0757.

Proctor, F.H, D.W. Hamilton, D.K Rutishauser, and G.F. Switzer, 2004: Meteorology and Wake Vortex Influence on American Airlines FL-587 Accident. NASA TM-2004-213018, April 2004, 58 pp.

Proctor, F.H., D.W. Hamilton, and G.F. Switzer, 2006: TASS Driven Algorithms for Wake Prediction. AIAA 2006-1073.

Proctor, F. H., D. A. Hinton, J. Han, D. G. Schowalter, and Y.-L. Lin, 1997: Two dimensional wake vortex simulations in the atmosphere: Preliminary sensitivity studies. AIAA. 97-0056.

Schowalter, D. G., D. S. DeCroix, G. F. Switzer, Y.-L. Lin, S. P. Arya, 1997: Toward three-dimensional modelling of a wake vortex pair in the turbulent planetary boundary layer. AIAA 97-0058.

Shen S., F. Ding, J. Han, Y.-L. Lin, S. P. Arya and F. H. Proctor, 1999: Numerical modeling studies of wake vortices: Real case simulations. AIAA 99-0754.

Switzer, G.F., and F.H. Proctor, 2000: Numerical Study of Wake Vortex Behavior in Turbulent Domains with Ambient Stratification, AIAA 2000-0755.

Switzer, G.F., F.H. Proctor, N.N. Ahmad, and F.M. Limon Duparcmeur, 2010: An Improved Wake Vortex Tracking Algorithm for Multiple Aircraft. AIAA 2010-7993.

OTHER

Ahmad, N.N., and F.H. Proctor, 2011: Simulation of Benchmark Cases with the Terminal Area Simulation System (TASS). AIAA 2011-1005, and reprinted in Research Disclosure Journal, Database Number 564036, April 2011, 14 pp.

Bacon, D. P., J. T. Dunn, M. R. Sands, and R. A. Sarma, 1991: Single burst nuclear cloud database, Volume 1 - Cloud rise and stabilization modeling. DNA-TR-90-36-V1, 151 pp.

Bacon, D. P., R. A. Sarma, and F. H. Proctor, 1986: Smoke injection into the atmosphere from large area fires. Rep. No. 86/1922, Science Applications International Corporation, McLean, VA. [Also in "Technical Papers Presented At The Defense Nuclear Agency Global Effects Review -- 7-9 April 1987, Volume I," Technical Report DASIAC-TN-87-35-V1, Defense Nuclear Agency, Washington, DC, 349-374.]

Bradford, M. L., D. J. Rusk, 1990: 3-D hierarchical real-time cloud modeling system phase II final report. BMO-TR-90-82, 241 pp.

Chen C. and J. A. Zak, 1989: A cloud model simulation of space shuttle exhaust clouds in different atmospheric conditions. NASA CR-4223, 136 pp.

Kaplan, M. L., and F. H. Proctor, 1991: The numerical simulation of dryline bulges forced by evanescent nonhydrostatic gravity waves. Preprints, 9th Conference on Numerical Weather Prediction, Amer. Meteor. Soc., pp. 384-387.

Kaplan, M. L., F. H. Proctor, J. W. Zack, V. M. Karyampudi, P. E. Price, M. D. Bousquet, and G. D. Coats, 1989: Mesoscale acid deposition modeling studies. NASA CR-4262, 120 pp.

Proctor, F. H., 1985: Application of radiative boundary conditions to nonhydrostatic primitive equation models. Preprints Seventh Conference on Numerical Weather Prediction, Amer. Meteor. Soc., pp. 291-298.

Rutishauser, D.K., 2011: Implementing scientific simulation codes tailored for vector architectures using custom configurable computing machines. Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 168 pp.

Wong, V. C., M. L. Kaplan, J. W. Zack, and F. H. Proctor, 1985: On an interaction between a parameterized regional model and a nonhydrostatic explicit-convection model. Preprints, Seventh Conference on Numerical Weather Prediction, Amer. Meteor. Soc., pp. 397-404.

Zak, J. A., 1987: Operational implications of a cloud model simulation of space shuttle exhaust clouds in different atmospheric conditions. NASA CR-4222, 80 pp.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-01-2014	Technical Memorandum	

4. TITLE AND SUBTITLE

Terminal Area Simulation System User's Guide -Version 10.0

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Switzer, George F.; Proctor, Fred H.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

411931.02.01.07.13.07

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-20345

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/TM-2014-218150

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 03
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Terminal Area Simulation System (TASS) is a three-dimensional, time-dependent, large eddy simulation model that has been developed for studies of wake vortex and weather hazards to aviation, along with other atmospheric turbulence, and cloud-scale weather phenomenology. This document describes the source code for TASS version 10.0 and provides users with needed documentation to run the model. The source code is programed in Fortran language and is formulated to take advantage of vector and efficient multi-processor scaling for execution on massively-parallel supercomputer clusters. The code contains different initialization modules allowing the study of aircraft wake vortex interaction with the atmosphere and ground, atmospheric turbulence, atmospheric boundary layers, precipitating convective clouds, hail storms, gust fronts, microburst windshear, supercell and mesoscale convective systems, tornadic storms, and ring vortices. The model is able to operate in either two- or three-dimensions with equations numerically formulated on a Cartesian grid. The primary output from the TASS is time-dependent domain fields generated by the prognostic equations and diagnosed variables. This document will enable a user to understand the general logic of TASS, and will show how to configure and initialize the model domain. Also described are the formats of the input and output files, as well as the parameters that control the input and output.

15. SUBJECT TERMS

FORTRAN; Simulation; Three Dimensional; Time Dependence

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	91	19b. TELEPHONE NUMBER *(Include area code)* (443) 757-5802

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18