HYPACT

HYbrid Particle And Concentration Transport Model

Version 1.2.0

User's Guide

August 20, 2001

by

Robert L. Walko

Craig J. Tremback

Martin J. Bell

*ASTER Division
Mission Research Corporation
P.O. Box 466
Fort Collins, CO 80525-0466

Abstract

HYPACT **-** the HYbrid PArticle and Concentration Transport Model - represents a state-of-the-art methodology for predicting the dispersion of air pollutants in 3-D, mesoscale, time dependent wind and turbulence fields. ***HYPACT*** allows assessment of the impact of one or multiple sources emitted into highly complex local weather regimes, including mountain/valley and complex terrain flows, land/sea breezes, urban areas, and other situations in which the traditional Gaussian-plume based models are known to fail.

HYPACT, developed by the *ASTER Division of Mission Research Corporation, represents the next generation of dispersion modeling systems. It combines the best features of grid-based Eulerian dispersion methodologies with Lagrangian particle dispersion modeling.

The ***HYPACT*** Lagrangian dispersion scheme is very flexible. Species can include gases, and a spectrum of aerosol sizes. The 2-D or 3-D wind and turbulence fields are provided by MRC/*ASTER's ***RAMS*** (Regional Atmospheric Modeling System) for forecast applications, or an observational network for diagnostic applications. A Lagrangian model is ideal for regimes in which the assumptions underlying Gaussian plume-based models are violated, such as highly sheared flows, recirculating coastal and mountain/valley wind systems, urban heat islands, plume fumigation and bifurcation.

Although the ***RAMS*** code can directly compute the dispersion of any number of "tracers" in a Eulerian framework, ***HYPACT*** has certain advantages because it combines in one code the best features of both the Lagrangian and Eulerian dispersion estimating methodologies. The advantage is greatest near a source region for tracers when the source is small and irresolvable on the Eulerian grid. A comparable Eulerian treatment would necessarily represent the source by a volume no smaller than one grid cell, and would immediately begin diffusing the tracer in adjacent cells. A Lagrangian approach, on the other hand, is fully capable of representing a source of any size, and of maintaining a concentrated, narrow plume downwind of the source until atmospheric dispersion dictates that it should broaden. In contrast, at large distances from the source, where the tracer plume is typically broad and well mixed, representation of the plume by Lagrangian particles can become inefficient due to the large number of particles required to achieve a smooth characterization of the plume. The hybrid Lagrangian and Eulerian approach used in ***HYPACT*** represents a tracer by Lagrangian particles near the source, but converts particles to Eulerian concentrations where appropriate at large distances downwind.

It can be run in Lagrangian particle, Eulerian concentration transport, or a hybrid mode. In the hybrid mode Lagrangian particles are used in the near source region. As the particles disperse to scales of the HYPACT grid, they are converted to concentrations for continued transportation in a Eulerian sense (they in effect become background concentrations to the remaining particles).

Sources in ***HYPACT*** can be single or multiple, instantaneous (explosive), continuous, or time varying for any of the specified species. Source geometry can include point, line area and volume sources of various orientations. The model domain can extend from an area as small as an industrial plant site to hundreds of kilometers (up to the size of the ***RAMS*** domain). The number of particles released is limited only by available memory and can exceed hundreds of thousands. Sources, species and emission scenarios can either be managed through the namelist, or through database files.

HYPACT is a modular FORTRAN 90 compliant code with new features being regularly added. Currently planned future modules include dry deposition, rain out, re-evaporation, plume rise, evaporative sources and chemical transformations. Further planned "house keeping" enhancements include the removal of inactive particles and memory reallocations, gridded source input (and background concentrations), various file input/output formats and history restarts. Other enhancements may include code parrallelization, an execution control GUI, and enhancement of visualization tools (***REVU*** and ***RINGI***). Collaborative development on any of these enhancements is welcome.

HYPACT User's Guide

Contents

About HYPACT

The ***HYPACT*** (HYbrid Particle And Concentration Transport) model code has been developed to simulate the motion of atmospheric tracers under the influence of atmospheric flow, including turbulence. In its current form, it is set up to be driven by meteorological output from the ***RAMS*** (Regional Atmospheric Modeling System), but the basic design allows the flexibility, with minor code changes, to provide a variety of other inputs as well, including observational data.

Although ***RAMS*** can itself perform a similar function by simulating the motions of any number of Eulerian tracer fields, ***HYPACT*** has certain advantages because of the ability to run the dispersion as a post-processing step and because of its Lagrangian formulation. The advantage is greatest near a source region for the tracer when the source is of small scale and unresolved on the Eulerian grid. ***RAMS*** would necessarily represent the source by a volume no smaller than one grid cell, and would immediately begin diffusing the tracer to the adjacent cells. A Lagrangian model, on the other hand, is capable of representing a source of any size, and of maintaining a concentrated, narrow tracer plume downwind of the source until atmospheric dispersion dictates that the plume should broaden. A unique feature of ***HYPACT*** is that once a plume of Lagrangian tracer particles becomes sufficiently broad downstream from a source region, it can be converted to a well-resolved concentration field and advected using an Eulerian formulation. This hybrid approach allows high particle emission rates to be specified for a source to achieve good plume resolution, without retaining all particles so far into the future that excessively large numbers of them accumulate. Tracer sources that are well resolved on a grid may alternatively be specified as gridded sources in ***HYPACT*** and transported entirely by a Eulerian formulation. ***HYPACT*** carries out Eulerian tracer prediction much faster than ***RAMS*** because it does not predict its own wind, pressure, temperature, moisture, or turbulence fields.

In order to run ***HYPACT*** from ***RAMS***, the latter must be run first to generate a series of output "analysis" files. These contain wind, potential temperature, and turbulent kinetic energy fields at the output times of the atmospheric simulation. ***HYPACT*** reads these data from the analysis files and interpolates them in time between file times (for each ***HYPACT*** timestep) for determining transport and diffusion. For Eulerian concentration prediction, ***HYPACT*** assigns the time-interpolated data to a series of nested grids identical with those of the ***RAMS*** simulation that generated the analysis files. Concentrations are predicted on these grids given user-specified initial values and source characteristics. For Lagrangian particle prediction, the time-interpolated wind and turbulence data are also interpolated in space to the location of each Lagrangian tracer particle. If nested grids are used in the ***RAMS*** simulation, the finest grid data defined at the location of each particle is used for the interpolation. The particles are moved through space and time based on the interpolated wind velocity plus a random motion scaled to the local turbulent intensity. A gravitational settling velocity may also be superimposed on the particle motion.

HYPACT sources, species and emission can either be defined in the ***HYPACT*** namelists, or in database files. This is designed to allow both quick and dirty runs, and a range of sophisticated scenarios from common data.

In addition to Eulerian concentration field outputs on the ***RAMS*** grids, concentration can be periodically diagnosed from the particle distribution on the ***RAMS*** grids (assuming a specified mass represented by each particle), and the combined Eulerian and Lagrangian concentration fields computed and output. The concentration is useful for contouring plume morphologies, determining instantaneous or time-integrated dosages at specified locations, computing chemical reaction rates between different tracer fields, etc. The raw particle files can also be output, allowing the user to diagnose the particle concentrations on a finer concentration grid. Concentration grid diagnosis and NCAR Graphics plotting features were included in the 0.2 beta version of ***HYPACT***, however, there are now included in ***REVU*** (***RAMS/HYPACT***

Evaluation and Visualization Utilities). For details on the use of this tool, please refer to the ***REVU*** User's Guide.

Similar to ***RAMS***, ***HYPACT*** is in a state of continual evolution. This document is the fourth edition of the ***HYPACT*** User's Guide, describing the model as of August 2001. ***HYPACT*** version 1.2.0 is FORTRAN 90 compliant and includes several ***RAMS*** modules and the ***RAMS*** and ***Utilities*** libraries. UNIX/Linux make commands and a system of make files and makefile includes are used for compiling the code. The make files are detailed in the section describing how to compile the ***HYPACT*** model.

Execution of ***HYPACT*** is controlled by a set of namelist variables usually contained in a file named *HYPACT_IN*. In order to operate ***HYPACT***, the user should be acquainted with the two primary means of setting parameters that control its functions. These are:

1. The various configuration parameters contained in *hcommons.h*, which define several array dimensions controlling the capacity of the model.
2. The variables in the *HYPACT_IN* namelist file.

The following sections of this document describes the function and use of each of the parameters and namelist variables, and how to set appropriate values for them.

New in HYPACT Version 1.2.0

New features since ***HYPACT*** version 1.1.0 include:

- Modified code structure and memory allocation. As a result, this version of ***HYPACT*** is far more robust than previous version. We strongly recommend users of previous versions upgrading to this version, as we will no longer support the older versions. Specifically, many of the routines now use "implicit none" statements and all c iralloc memory allocations are now done with FORTRAN 90 allocations. This has allowed us to do bounds checking on arrays has which in turn has resulted in the cleaning up of a number of known and unknown bugs.
- Several long time bug fixes and ***RAMS*** / ***HYPACT*** code consistency checks.
- A fix to the hybrid conversion routine, ensuring stable plumes of Lagrangian particles advecting (about) 45 degrees from an north, south, east or west directions do not prematurely get converted to Eulerian concentrations.
- Accepts 1 argument, -f, for pointing to non-standard *HYPACT_IN* file names, i.e.

```
hypact-1.2.0 -f <namelist file>
```

Installing and Running HYPACT

The ***HYPACT*** version 1.2.0 code is FORTRAN 90 compliant. ***HYPACT*** has been compiled and run on SGI, HP, IBM and PC-Linux workstations. If you are compiling on other platforms we would like to know what compiler options you use. To convey this information, please contact:

rams-support@aster.com

Keeping Up To Date with HYPACT

The latest version of ***HYPACT*** is available from:

http://www.aster.com/hypact.shtml

Patches will be made available at:

http://www.aster.com/hypact-1.2.0/patch.shtml

as they are made. Notification of new patches will be sent to the rams-users mailing list and are available for viewing on the Announce archive:

http://www.aster.com/lists/announce

You can apply to join the rams-users mailing list at:

http://www.aster.com/lists/index.shtml

As always, we appreciate any bug reports and compilation options that are different or not offered in the distributed make files. Please see the notes on reporting bugs before seeking help.

Installing HYPACT

This section describes the compilation of ***HYPACT*** version 1.2.0. This version is compatible with ***RAMS*** version 4.3.0, ***REVU*** version 2.3.1 and ***UTILS*** version 1.0.0. The reader should also check the release notes in the *README_HYPACT-1.2.0* file. Note that all MRC/*ASTER software should be installed under a common "rams" directory (usually this directory is named *rams*, but it can be anything you wish). This directory is referred to as the *RAMSROOT* in this user manual.

Downloading the Latest Version of HYPACT

1. Obtain the latest version of the software by downloading the UNIX gzipped tar file *hypact-1.2.0.tar.gz* from the web address:

   http://www.aster.com/hypact-1.2.0/hypact-1.2.0.tar.gz

2. Move the tar file into *RAMSROOT*:

```
mv hypact-1.2.0.tar.gz RAMSROOT
```

3. Unpack the contents of the tar file by ***either***:

```
gunzip -c hypact-1.2.0.tar.gz | tar -xf -
```

   or, if you are on a Linux machine:

```
tar -zxf hypact-1.2.0.tar.gz
```

   Upon completion of this step you should find the latest versions of the *README* file, source code, make files, and configuration files in the *src*, *bin* and *run* directories in *RAMSROOT*:

```
RAMSROOT/bin/README_HYPACT-1.2.0    - build and install files
           ./Make.hypact-1.2.0
           ./dep_hypact-1.2.0.mk
           ./Makefile-hypact
           ./include.mk-std
           ./check
           ./versions
           ./Copyright

RAMSROOT/run/HYPACT_IN              - configuration files
           ./HYPACT_IN-files
           ./species
           ./sources
           ./emissions

RAMSROOT/src/hypact/1.2.0/model/[source]    - source code
                        ./model/modules/[modules]
                        ./include/[includes]
```

Downloading the Latest Patch for HYPACT

1. Obtain the latest version of the software, by ***either*** downloading the UNIX gzipped tar file *hypact-1.2.0-patch.tar.gz* from the web address:

   `http://www.aster.com/hypact-1.2.0/hypact-1.2.0-patch.tar.gz`

   - Make a temporary directory and move the tar file in to that directory:

     ```
     mkdir tmp
     mv hypact-1.2.0-patch.tar.gz tmp
     ```

   - Unpack the contents of the patch tar file by ***ether***:

     ```
     gunzip -c hypact-1.2.0-patch.tar.gz | tar -xf -
     ```

     or, if you are on a Linux machine:

     ```
     tar -zxf hypact-1.2.0-patch.tar.gz
     ```

   or, download patched modules individually from:

   `http://www.aster.com/hypact-1.2.0/patch.shtml`

2. Replace the modules in your distribution with those new versions contained in the patch, noting that since the source comes with read only permissions, you will need to modify the permissions of the module in your *RAMSROOT/src* directory before replacing it with the patch version. For example:

   ```
   chmod u+w RAMSROOT/src/hypact/1.2.0/model/hsource.f90
   mv src/hypact/1.2.0/model/hsource.f90 RAMSROOT/src/hypact/1.2.0/model
   ```

3. You may then wish to remove read permissions from the new module:

   ```
   chmod u-w RAMSROOT/src/hypact/1.2.0/model/hsource.f90
   ```

Preparing to Compile HYPACT

Before you compile the software:

1. Go to the bin directory:

   ```
   cd RAMSROOT/bin
   ```

2. If you have not done so already, move the Makefile-hypact to Makefile:

   ```
   mv Makefile-hypact Makefile
   ```

3. ***Either***, if this is your first time using *include.mk*, move the *include.mk-std* to *include.mk* and modify it (with vi, for example) to suit your system:

   ```
   mv include.mk-std include.mk
   vi include.mk
   ```

   include.mk contains all the make environment variables that a user might need to change in order to compile the code on their machine. It is included in all of the make files using the include command.

   FIRST TIME USERS MUST CUSTOMIZE THIS FILE BEFORE THEY ATTEMPING TO MAKE.

   By default, *include.mk-std* does not have the compiler flags for any machine type switched on.

   Or, if you already have a copy of *include.mk* that you have modified to suit your system, check your *include.mk* with the new *include.mk-std* for software system changes in *include.mk-std* and make those changes to your *include.mk* (with vi, for example):

   ```
   diff include.mk include.mk-std
   vi include.mk
   ```

- *dep_hypact-1.2.0.mk* contains all the dependencies within the ***HYPACT*** distribution. This means that if a file such as *hcommons.h* were updated, all those modules that use this file would be recompiled (noting that *hcommons.h* is not itself compiled, but included in whatever modules require it when they are compiled). All include (.h) and module (.mod) files are treated in the same way.
- If you have multiple versions of ***RAMS***, ***HYPACT***, ***REVU*** or the ***UTILS*** library, the version built by default make command will be that indicated near the top of the *include.mk* file. The set of version numbers in the *include.mk* file should match a set of version numbers in the *versions* file.
- All make commands will run the shell script *check* which outputs some advice if it finds a version mismatch (it does not stop the make command from completing, although you may then get compilation errors). *check* compares the versions indicated in the *include.mk* file with the list of compatible versions in the *versions* file and the versions installed under *RAMSROOT* on your machine. If the check script fails on your machine you can remove the "check" dependency from the "all" target in each of the make files.
- If the include command does not appear to work on your machine, try replacing the include command line in each of the make files with the contents of the include file. Alternatively, you can download "GNU make" and use that instead of your platform version. "GNU make" is available from the URL:

  http://www.gnu.org/software/make/make.html
- All readme, make and dependency files are distributed with a version number appended. You can optionally remove this from the make file names (do not remove them from the dependency file names).

Compiling HYPACT

- To compile the software, use ***either***:

  ```
  Make -f Make.hypact-1.2.0
  ```

  or, to use the global make file *Makefile*, enter:

  ```
  make hypact
  ```

  This should produce the ***HYPACT*** archive library *hypact-1.2.0.a*, the ***HYPACT*** executable *hypact-1.2.0,* and a link from *hypact* to *hypact-1.2.0.*

  or, to use the global make file *Makefile* to update all the executables in your *bin* directory:

  ```
  make
  ```

This will ensure that all the executables are up to date, noting that there are a number of source code cross dependencies in the software system.

- To recompile ***HYPACT*** when any of the ***REVU***, ***RAMS*** or ***UTILS*** library modules are updated, repeat the make command. If you do not "clean" your build, this will update the executable, recompiling only those codes that have been updated, or that depend on codes that have been updated.

Installing HYPACT

- *Make.hypact-1.2.0* has an "install" target that will install (as a symbolic link) the executable to the *run* and *test* directories:

  ```
  make -f Make.hypact-1.2.0 install
  ```

  You can modify *Make.hypact-1.2.0* if you wish to install to alternative locations. You can also manually link to the executables (does the same as above):

  ```
  ln -s ../bin/hypact .
  ```

  On some machines you may need to make a physical copy of the executable in your test directory:

  ```
  cp ../bin/ hypact-1.2.0 hypact
  ```

  Don't forget to update copied executables when you remake in the bin directory otherwise your changes will not appear in your runs.

Cleaning the REVU Installation

- *Make.hypact-1.2.0* also has a "clean" target that will remove built components. To clean out the compiled libraries and executable:

  ```
  make -f Make.hypact-1.2.0 clean
  ```

  This should remove the ***HYPACT*** archive library *hypact-1.2.0.a*, the ***HYPACT*** executable *hypact-1.2.0* and the link from *hypact* to *hypact-1.2.0.*

- *Makefile* also contains a "clean_all" target. Entering the following will result in all libraries and executables being removed (not just the ***HYPACT*** components):

  ```
  make clean_all
  ```

Running HYPACT

1. Check the settings of the variables in the parameter statements in *hcommons.h*, and change if necessary. If you change any of these setting you will need to recompile ***HYPACT***.
2. Run ***RAMS*** with the following namelist settings:
   - KWRITE = 1 to write the eddy diffusion coefficient parameters to the ***RAMS*** analysis files for Eulerian diffusion.
   - IDIFFK ≠ 2 or 3 since these do not produce TKE's for Lagrangian particle dispersion.
3. Copy and configure *HYPACT_IN* for the simulation to your working directory.
4. Run the ***HYPACT*** executable in your working directory using ***either***:

   ```
   hypact-1.2.0
   ```

   or, if using a *HYPACT_IN* file with a non-standard name:

   ```
   hypact-1.2.0 –f HYPACT_IN-non-standard
   ```

Visualizing HYPACT Results

1. Download the latest ***HYPACT*** compatible version of ***REVU*** from:

   `http://www.aster.com/aster/revu.shtml`

2. Install and run noting the new field names in the ***REVU*** documentation.

HYPACT Test Runs

****** Coming Soon ******

Reporting Bugs in HYPACT

Before you seek help:

1. Ensure that you have the latest version of the software. Check for patches at:

   ```
   http://www.aster.com/hypact-1.2.0/patch.shtml (HYPACT)
   http://www.aster.com/rams-4.3.0/patch.shtml   (RAMS)
   http://www.aster.com/revu-2.3.1/patch.shtml   (REVU)
   http://www.aster.com/utils-1.0.0/patch.shtml  (UTILS)
   ```

   Notification of new patches will be sent to the rams-users mailing list and are available for viewing on the Announce archive:

   `http://www.aster.com/lists/announce`

   You can apply to join the rams-users mailing list at:

   `http://www.aster.com/lists/index.shtml`

2. If you are still having problems:

   - Note your machine type, operating system (and version) and compiler (and version if possible).
   - Copy the screen output to a file. Try using the script command:

     ```
     script -a <file>   (to start the script shell and direct the output)
     hypact-1.2.0       (run programs)
     exit               (exit script shell)
     ```

     Or redirect both standard output and error to a file. For example, running rams:

     ```
     hypact-1.2.0 1>&2 <file>  (Korn and Bash shells)
     hypact-1.2.0 >>& <file>   (C shell)
     ```

   - Note what configuration or make files were used.
   - Send to all of this plus any other supporting information to:

     rams-support@aster.com

Future Developments in HYPACT

Future upgrades may include:

- Dry Deposition.
- Rain Out.
- Re-evaporation.
- Plume Rise.
- Evaporative Sources.
- Ground dust emissions.
- Gridded Source Input (and background concentrations).
- Chemical Transformations.
- Removal of Inactive Particles.
- GRIB file output.
- History Restarts.
- Execution Control GUI.
- Code Parrallelization (in-line with ***RAMS*** version 5.0).

Future visualization tools may include:

- Concentration grid post process tool and viewer (in ***REVU***) and ***REVU*** GUI (***RINGI***).

Further development ideas and collaboration, questions and bugs reports (please save relevant ***RAMS*** analysis and *HYPACT_IN* files) are welcome and can be directed to:

rams-support@aster.com

HYPACT Configuration Parameters

All ***HYPACT*** parameters are set in *hcommons.h*.

nvar3d **nvar2d** **nvarsd** integer	Specifies the number of 3-D atmospheric, 2-D, and 3-D soil arrays, respectively, that are read from ***RAMS*** analysis files. They are multiplied by the respective sizes of each array type and summed to determine the amount of memory to allocate in ***HYPACT*** for containing ***RAMS*** output fields. Values for these parameters should only be changed if one is modifying the ***HYPACT*** code to read in more ***RAMS*** fields.
maxg integer	Used to dimension several arrays in the *hcommons.h* common block file. **maxg** must be greater than or equal to the number of grids used in a ***RAMS*** simulation used to drive ***HYPACT***.
maxi **maxj** **maxk** integer	Used to dimension several arrays in the *hcommons.h* common block file. They must be greater than or equal to the largest number of grid points spanning the x-direction (east to west), y-direction (south to north) and z-direction (vertical), respectively, of any grid used in a ***RAMS*** simulation used to drive ***HYPACT***.
maxhi **maxhj** **maxhk** integer	Used to dimension several arrays in the *hcommons.h* common block file. They specify upper bounds on the number of grid points spanning the i, j, and k directions, respectively, of the grid defined in ***HYPACT*** (*h-grid).* The *h-grid* is a composite of all nested grids used in a ***RAMS*** simulation used to drive ***HYPACT***. It has variable spatial resolution according to the mesh sizes, overall dimensions, and locations of all grids in ***RAMS***. The *h-grid's* mesh size is always at least as fine as, and often finer than, the finest ***RAMS*** grid mesh size at any given location. The required number of points spanning each direction of the *h-grid* is computed by a complicated algorithm, but is usually less than the sum over all ***RAMS*** grids of the numbers of points spanning each corresponding direction of the ***RAMS*** grids. Thus, if for example ***RAMS*** uses 3 grids each of (i,j,k) dimensions (30,40,50), the *h-grid* would likely have dimensions smaller than (90,120,150). These parameters need not match the exact size of the *h-grid*; they only have to be large enough to contain it.
maxfiles integer	Used to dimension several arrays in the code. **maxfiles** must be greater than or equal to the largest number of ***RAMS*** 'analysis' files used in a ***HYPACT*** simulation that have the path and filename prefix given by **hyppref** (not just those within the defined ***HYPACT*** simulation time).
maxspec integer	Used to dimension several arrays in the *hcommons.h* common block file and the code. **maxspec** must be greater than or equal to the largest number of species to be used in a ***HYPACT*** simulation.
maxsrc integer	Used to dimension several arrays in the *hcommons.h* common block file and the code. **maxsrc** must be greater than or equal to the largest number of sources to be used in a ***HYPACT*** simulation.
maxem integer	Used to dimension several arrays in the *hcommons.h* common block file and the code. **maxem** must be greater than or equal to the largest number of emissions to be used in a ***HYPACT*** simulation. The number of emissions used is the tally of emissions turned on by **$EMISSIONS** namelist variable **iemit**.

polypts integer	Used to dimension several arrays in the *hcommons.h* common block file. **polypts** must be greater than or equal to the largest number of vertices used to define any polygon emission source in a ***HYPACT*** simulation. The number of vertices used for each polygon is specified by the **$SOURCES** namelist variables **polylat** and **polylon**.
maxent integer	Used to dimension one array in the *hcommons.h* common block file, and specifies the size of a table of normally-distributed numbers used in ***HYPACT***. A value of approximately 1000 provides adequate density in the table.

HYPACT Namelist and Database Files

The user has the flexibility of emitting particles and concentrations from different locations, at different times, at different rates, and with different source region characteristics. Each particle carries with it a label indicating the source from which it was emitted, so the particles can be identified accordingly throughout their journey.

A database approach is used in the specification of species and sources in the *HYPACT_IN* namelist file. As such, any number of species and sources may be individually defined in the **$SPECIES** and **$SOURCES** namelists (within the limits of the ***HYPACT*** parameter settings). Any combinations of species and sources can then be specified in conjunction with emission data (rate, duration, etc) in the **$EMISSIONS** namelist. Thus the *HYPACT_IN* namelist file can contain a comprehensive species and sources datasets, with the **$EMISSIONS** namelist using only that data required for the current run. In addition to this flexibility, any number of emissions (within the limits of the ***HYPACT*** parameter settings) can be specified in the **$EMISSIONS** namelist.

Alternatively, the species, source and emissions specifications included in the namelists can be carried in separate 'database' files. If using species, source and emissions database files only minimal number of parameters are required in the **$SPECIES**, **$SOURCES** and **$EMISSIONS** namelists. The format for these files is discussed in the following sections.

Two other namelists **$GENERAL** and **$OUTPUT** control the other aspects of running the model.

$GENERAL Namelist

ihyprun integer	The ***HYPACT*** run type. • If set to 1, a source analysis only will be performed **-** the ***HYPACT*** run is stopped before the first timestep proceeds with advection and diffusion. The initial output files will contain the initial emission field, providing a useful way for the user to check their work. Note that a summary of each source is also printed to standard output prior to the first timestep. • If set to 2, the dispersion run is made (similar to the ***RAMS*** 'INITIAL' start).
metpref character	The UNIX path name and prefix of the names of ***RAMS*** analysis files to be read and used to drive the ***HYPACT*** simulation (up to 80 characters). Note that the path name is optional and can be either relative or absolute (no path is equivalent to './'). The analysis files to be used must all exist in this same directory, and must span a range of simulation time that brackets the range of time over which a ***HYPACT*** simulation is to be run.
dtpart real	The length of the discrete timestep in seconds used to update particle velocities and positions. Its value should be carefully chosen based on required speed and accuracy of the ***HYPACT*** run. It would be desirable for each particle to move no more than the dimension of a grid cell (on which the atmospheric data are represented), and considerably less if practical, in a single timestep so that the maximum amount of detail contained in the atmospheric data is utilized. This may at times require a very small value of **dtpart,** such that a large number of individual timesteps must be performed to run ***HYPACT*** for a given span of simulation time. If this runs too slowly, and/or if a lower level of accuracy is acceptable, **dtpart** may be longer. An estimate for an appropriate value of **dtpart** can be obtained by dividing grid cell dimensions by atmospheric wind velocity components.

maxpart integer	The total number of particles that can be used in a ***HYPACT*** run. An appropriate number can vary widely from one application to another. If one only wants a general idea of which direction particles are traveling from a source, fewer than 1000 particles may be totally adequate. If concentrations are to be computed from the particle distribution, it is desirable to have as many particles as practical to improve the accuracy of the computation. Tens of thousands of particles might be used in this case, or even more if concentrations far from a dense particle plume are sought. Of course, ***HYPACT*** runs slower when many particles are activated. Note that **maxpart** does not specify the number of particles actually emitted (this is done by **numparts**, described in the **$EMISSIONS** namelist) but is the maximum number allowable from computer memory considerations.
freqavg real	The time interval in seconds between successive updates of the time average applied to input meteorological data to define it at times close to the current simulation time of the particles in ***HYPACT***. The meteorological data (for example, the output analysis files from ***RAMS***) is defined for discrete points in time, spaced usually at regular intervals such as an hour. Interpolation between consecutive times of the data is performed so that particles use winds and turbulence parameters close to the correct time. Since frequent averaging takes time and is unnecessary **freqavg** is usually set to a value many times longer than the ***HYPACT*** timestep **dtpart**, but several times shorter than the interval between input data files. Must be less than or equal to the ***RAMS*** analysis file frequency, and greater than of equal to **dtpart**.
ihturb integer	A flag specifying whether a turbulent component is to be added to the mean resolved atmospheric motion for advecting particles. • If set the 0, no turbulence will be included, and the ***HYPACT*** Lagrangian particle model operates purely as a trajectory model. • If set to 1, atmospheric turbulent intensity is evaluated and used to derive a turbulent component of motion for each particle. This allows the effects of unresolved atmospheric motions on particle transport and dispersion to be statistically accounted for.
iadvord integer	The order of the advection operator to be used in transporting Eulerian concentration fields. The choices are 2 for second order and 6 for sixth order.
hybfreq real	When an emission type (**type**) is specified as 'hyb' the Lagrangian particles emitted for the source are converted to Eulerian concentrations at some point downwind (when the computed dispersal of particles is sufficient). This conversion is handled by releasing the particles in clusters The spread of each cluster is then checked every **hybfreq** seconds, and if the following criteria is satisfied the particles are converted to Eulerian concentrations. **hybfreq** should be set to within a few times **dtpart**. If **hybfreq** is less than **dtpart**, clusters will be checked every ***HYPACT*** timestep. The criterion is a measure of spread between the particles compared to the specified horizontal and vertical distances. The spread is taken as six times the standard deviation of cluster particle locations done in each of x, y and z directions. This is then checked against **hybhoriz** (x and y) and **hybvert** (z), and the conversion made only if the spread in all three dimensions is greater than the **hybhoriz** and **hybvert** criteria.
ihybpart integer	The number of particles emitted per cluster. Different clusters will be allocated to each source so that no cluster will contain a variety of species. **ihybpart** should be set that so that a cluster of particles does not extend over more than several ***HYPACT*** timesteps.

ihybmin integer	The minimum particles per cluster. If there are less than **ihybmin** particles in a cluster the cluster will be converted to Eulerian concentrations regardless of the spread. This value must be less than **ihybpart** else all particles will be converted when their respective clusters are full. Conversion of a cluster that a source is in the process of filling will not take place until that cluster is full, or the source's emission is complete. A recommended setting for **ihybmin** is 10% of **ihybpart**.
hybhoriz real	The horizontal length scale defined as fractional horizontal grid spacing. With a spread of six standard deviations and **hybhoriz** and **hybvert** is equal to 1, cluster conversions are made approximately when the particles have dispersed over at least the entire grid volume in each dimension. Settings of **hybhoriz** and **hybvert** should therefore be in the range 1 to 3.
hybvert real	The vertical length scale defined as fractional vertical grid spacing or the distance in meters if **hybvert** is less than 0. With a spread of six standard deviations and **hybhoriz** and **hybvert** is equal to 1, cluster conversions are made approximately when the particles have dispersed over at least the entire grid volume in each dimension. Settings of **hybhoriz** and **hybvert** should therefore be in the range 1 to 3.

$OUTPUT Namelist

Almost all output is controlled by the **$OUTPUT** namelist, with the sole exception that the **ioutspec** variable in the **$SPECIES** namelist controls whether individual species are included in the output.

hyppref character	The UNIX path name and prefix for the ***HYPACT*** output files generated during the simulation (up to 80 characters). Note that the path is optional and can be either relative or absolute (no path is equivalent to './'). The time interval between the writing of successive files in the set is specified by the namelist variable **hypfrep**.
hypfreq real	The time interval in seconds between successive writing of ***HYPACT*** output files. The settings of **ipartout**, **ieulout** and **ilagout** specify what variables are output, and **irgrid** specifies the ***RAMS*** grids that are included in the output. Output file names and formats are described separately following the namelist documentation.
ipartout integer	Particle file output flag. **ipartout** set to 1 produces particle files every **hypfreq** seconds provided at least one element of **type** is equal to 'lag', 'hyb' or 'both'.
ieulout **ilagout** integer	Output concentrations are controlled with the following, noting that there is one concentration field for each species; • **ieulout** gives the Eulerian concentrations. These fields will also contain the concentrations produced by the conversion of particle clusters in hybrid mode (**$EMISSIONS** namelist parameter **type** set to 'hyb'). • **ilagout** gives the concentration fields computed by converting all particles to Eulerian concentrations (the particles still remain in the Lagrangian field). For each: • 0 gives no output. • 1 gives the concentrations at the end the current timestep. • 2 gives the average concentrations over **avgtime** seconds prior to the file output time (determined by **hypfreq**). • 3 gives both the current and average concentration output. The hybrid concentrations (when an emission type (**type**) is specified as 'hyb') are the sum of the Lagrangian particles yet to be converted and Eulerian concentrations from converted clusters for each species. This summation is handled in the post processing with ***REVU***.
imetout integer	Meteorology output control flag. **imetout** set to 1 outputs the interpolated ***RAMS*** meteorology as used by ***HYPACT*** at the output time. The meteorology is included in the concentration file output.
avgtime real	Averaging time in seconds over which concentration field averaging is done before each output time. Averaging can be done for any species on the ***RAMS*** grid. It cannot be done for individual emission sources unless each emission source emits a different species, and cannot be done for the particle files. **avgtime** must be less than or equal to the **$GENERAL** namelist parameter **hypfreq**.

ioutfmt integer	Output file format: • If set to 1, VFILE binary files are output.
irgrid integer	***HYPACT*** output is limited to the grids specified by **irgrid**. • If set to 0, concentrations and particles are output on all grids. • If set to greater than 0, concentrations are output for that grid only. • If set to less than 0, concentrations are output for all grids less than the absolute value of **irgrid**. In the latter two cases, particles in the output file include all those that reside within the bounds of the specified **irgrid**.

$SPECIES Namelist

specfile character	The UNIX path and filename containing the species database (up to 80 characters). An example file and the file format follows this table. Note that the path is optional and can be either relative or absolute (no path is equivalent to './'). If **specfile** is set to 'none' the species database is read from the following namelist parameters. If **specfile** points to a database file, with the exception of **ioutspec**, the remaining **$SPECIES** namelist parameters are overwritten.
specname character array[†]	A multi-valued parameter specifying the name of each species (up to 30 characters long). **specname** can include standard NCAR Graphics sub and superscripting code. This information is stored in the output 'SPEC' file, and so it is useful for later plotting with NCAR Graphics. 'N' denotes the beginning of normal text, 'B' subscripted text and 'S' superscripted text (all ":" delimited). It is safest to start and end with an 'N', for example, ':N:CO:B:2:N:' results in CO_2.
Wgtmol real array[†]	A multi-valued parameter specifying the molecular weight of each species in grams per mole. **wgtmol** is used to calculate Eulerian concentrations in parts per trillion if **units** is in nanograms (ng), parts per million (ppm) if **units** is in micrograms (ug) and parts per thousand if **units** is in milligrams (mg).
Units character array[†]	A multi-valued parameter specifying the mass units applied to the emission rate and ratio (up to 30 characters long). **units** can be set to ng (nanograms), ug (micrograms) or mg (milligrams). The units set here should be consistent with those used for the **$EMISSIONS** namelist parameters, **rate** and **ratio**.
Ihfall integer array[†]	A flag specifying whether particles in ***HYPACT*** are to be caused by gravity to settle relative to the local flow in which they are embedded. A value of 0 indicates no settling, while a value of 1 activates the settling. The parameters **szmin**, **szmax**, and **szpwr** below control the settling velocity for particles from each source.
Szmin real array[†]	A multi-valued parameter specifying a minimum particle diameter in meters emitted from each source region. Its only use is in determining gravitational settling speed, and is ignored if namelist variable **ihfall** is set to 0.
Szmax real array[†]	A multi-valued parameter corresponding to namelist variable **szmin**, but applying to the maximum particle diameter from each source region. Currently, only a Stoke's drag formula is implemented in ***HYPACT***, which is relevant for diameters up to about than 80×10^{-6} m.
Szpwr real array[†]	A multi-valued parameter relating closely to **szmin** and **szmax**, and is used in computing gravitational settling speed. It specifies a particle size distribution, effectively weighting the particles toward the larger or smaller sizes. **szpwr** specifies the power of the diameter which is linearly (uniformly) distributed between the specified size limits. For example, if **szpwr** is set to 1, all diameters between the limits are emitted with equal probability. If **szpwr** is set to 3, all cubes (third power) of particle diameters, or equivalently all volumes or masses, are emitted with equal probability between the size limits. **szpwr** may be any non-negative number.
ioutspec integer array[†]	A multi-valued parameter specifying what species are to be included in the output. To output the species, set **ioutspec** to 1.

[†] Each array must contain entries for each species.

Species Database File

The following illustrates some species in the database file format:

```
1,3
TRS,60,ug,0,1.,1.,1.
:N:CO:B:2:N:,44,ug,0,1.,1.,1.
XXT,98,ug,0,1.,1.,1.
```

Where the header line contains file format version and the number of species in the file with each subsequent line containing all the information for each species in comma delimited format (order is important):

- **specname**, **wgtmol**, **units**, **ihfall**, **szpwr**, **szmin**, **szmax**

Note that the **$SPECIES** namelist variable **ioutspec** is not included in the species database file, and should be included in the **$SPECIES** namelist. If it is not, the default is to output for all species.

$SOURCES Namelist

srcfile character	The UNIX path and filename containing the source database (up to 80 characters). An example file and the file format follows this table. Note that the path is optional and can be either relative or absolute (no path is equivalent to './'). If **srcfile** is set to 'none' the source database is read from the following namelist parameters. If **srcfile** points to a database file the remaining **$SOURCES** namelist parameters are overwritten.
srcname character array‡	A multi-valued parameter specifying the name of each source (up to 30 characters long).
shape character array‡	A multi-valued parameter specifying the horizontal cross-sectional shape of each source. **shape** can be set to 'point', 'rectangle', 'triangle', 'polygon', or 'ellipse' (these can be abbreviated to the first three characters). • If a point source is specified, namelist variables **srcx** and **srcy** are used to determine the horizontal location of the source, and **srcz** the height of the source (this is converted to a rectangular source with no dimensions in ***HYPACT***). • If a triangular of irregular polygon shape is specified, the namelist variables **polylat** and **polylon** are used to specify the horizontal coordinates of the vertices, **srcx**, **srcy** the center of the polygon (see the notes on **polylat** and **polylon** for the relevance of this**)**, and **srcz** and **zsize** the height and thickness of the source. • If a rectangular or elliptical shape is specified, namelist variables **srcx**, **srcy**, **xsize**, and **ysize** are used to determine the horizontal location and size of the source, and **srcz** and **zsize** the height and thickness of the source. **rotation** can also be specified to rotate rectangular or elliptical sources in the horizontal plane. • If a gridded shape is specified, the emissions are obtained from the file named in **gridfile.** (not implemented) Note that the center location of each source, and the entire source region, should lie within the volume over which the input meteorological fields are defined (i.e., the coarsest grid of the ***RAMS*** domain). Any location outside this range places the source outside the model domain, and emissions there will remain stationary in ***HYPACT***.
Srcx **srcy** real array‡	Multi-valued parameters specifying the horizontal coordinates of the center of each source region. **srcx** and **srcy** are used only for point, rectangular, polygonal and elliptical sources specified by **shape,** and indicate the geographic longitude and latitude, respectively, in degrees.
Srcz real array‡	A multi-valued parameter specifying the height above the ground in meters, and applies for all source types.

Polylat
polylon
real array[†]
	Multi-valued two-dimensional parameters specifying the geographic vertex latitude and longitude coordinates for triangular and irregular polygon shaped sources specified by **shape**. The indices on the 2-D array are vertex and source. The vertex index must always be equal to 1 (first array element), while the source index refers to the source number. For irregular polygon sources, the vertices must be defined in an anti-clockwise order - the first and last coordinates are joined to enclose the polygon. For example, the following describes a triangular source for source 2, and a four-sided polygon for source 3:
	`polylat(1,2) =   40.075,  40.076,  40.071,`
	`polylon(1,2) = -104.721,-104.723,-104.722,`
	`polylat(1,3) =   40.075,  40.075,  40.071,  40.071,`
	`polylon(1,3) = -104.721,-104.723,-104.723,-104.721,`
	Polygonal sources were conceived to facilitate simpler specification of irregularly shaped sources, such as urban emission boundaries. Polygonal sources are broken up into an array of triangular sources in the ***HYPACT*** code. Each defined polygon must be sufficiently simple so that none of the triangles overlap, and no part of a triangle lies outside the polygon. Tests for these limitations are made in the code. In breaking up a polygon, the vertices of each triangle are initially taken as the two points of one side of the polygon, and a point somewhere in the center of the polygon (as defined by **srcx** and **srcy**). If any triangle fails a test, the central point is moved in the code, and the triangles redefined. If the code generated triangles continue to fail a test, the program is stopped and the user is required to reassess the configuration of their sources.

Xsize **ysize** real array[†]	Multi-valued parameters specifying the x and y direction dimensions in meters of rectangular and elliptical sources specified by **shape**. • If rectangular, **xsize** and **ysize** are the dimensions respectively. • If elliptical, **xsize** and **ysize** represent the lengths of the major and minor axes respectively.
Zsize real array[†]	Similar to **xsize** and **ysize**, but specifying the vertical dimension of the source in meters. If **zsize** is greater than zero, half of the source will be below the specified source center height **srcz**. Thus, the latter should be sufficiently high that no part of the source is underground (i.e., at least half the value of **srcz**).
Rotation real array[†]	A multi-valued variable indicating a rotation angle (in degrees) in a clockwise manner about a vertical axis of a source. This allows basic shapes such as rectangle or ellipse to be first indicated with principal axes parallel to the x- and y-axes of the ***HYPACT*** grid, and then for those shapes to be re-oriented in any direction desired. A positive angle causes a clockwise rotation as viewed from above.

[†] Each array must contain entries for each source, noting the flexibility demonstrated in the example below.

Source Namelist Examples

The following gives an example for each type of source. Note that different parameters apply in different ways to the sources, and some placeholders may be required, although they will not get used. There are two recommended ways to specify sources - either by using placeholders:

```
srcname='Mill_1','Power_1','Power_2','Ponds','Mill_2',

shape='rec','tri','poly','ell','poi',

srcy=40.2,0.,39.7,40., 40.2,
srcx=-104.2,0.,-82.7,-105.,-105.2,
srcz=50.,100.,100.,5.,50.,

xsize=100.,0.,0.,8000.,0.,
ysize=100.,0.,0.,8000.,0.,
zsize=15.,10.,100.,10.,0.,

polylat(1,2)=40.075,40.076,40.071,
polylon(1,2)=-104.721,-104.723,-104.722,
polylat(1,3)=40.075,40.075,40.071,40.071,
polylon(1,3)=-104.721,-104.723,-104.724,-104.720,

rotation=45.,0.,0.,40.,0.,
```

Note the placeholders:

- The second values of srcy and srcx are not used for the triangular source.
- The second, third and fifth values of xsize, ysize and rotation are not used for the triangular, polygonal and point sources.

Note also that only the second and third series of polylat and polylon points (for the triangular and polygonal sources) are required. In this case the two dimensional namelist variables must contain the relevant indices (the second index refers to the source number).

Or alternatively, to remove the need for placeholders - each source can be specified with the relevant indices on the namelist variables:

```
srcname(1)='Mill_1',
shape(1)='rec',
srcy(1)=40.2,
srcx(1)=-104.2,
srcz(1)=50.,
xsize(1)=100.,
ysize(1)=100.,
zsize(1)=15.,
rotation(1)=45.,

srcname(2)='Power_1',
shape(2)='tri',
srcz(2)=100.,
polylat(1,2)=40.075,40.076,40.071,
polylon(1,2)=-104.721,-104.723,-104.722,
zsize(2)=10.,
srcname(3)='Power_2',
shape(3)='poly',
srcy(3)=39.7,
srcx(3)=-82.7,
srcz(3)=100.,
polylat(1,3)=40.075,40.075,40.071,40.071,
polylon(1,3)=-104.721,-104.723,-104.724,-104.720,
zsize(3)=100.,

srcname(4)='Ponds',
shape(4)='ell',
srcy(4)=40.,
srcx(4)=-105.,
srcz(4)=5.,
xsize(4)=8000.,
ysize(4)=8000.,
zsize(4)=0.,
rotation(4)=40.,

srcname(5)='Mill_2',
shape(5)='poi',
srcy(5)=40.2,
srcx(5)=-105.2,
srcz(5)=5.,
```

Source Database File

The following illustrates the above sources in the database file format:

```
1,5
Mill_1,rec,40.,-83,50.,1.,1.,1.,0.
Power_1,tri,40.,-83,100.,40.075,-104.721,40.075,-104.723 \
               ,40.071,-104.722,0.
Power_2,poly,40.,-83,100.,4,40.075,-104.721,40.075,-104.723 \
                ,40.071,-104.724,40.071,-104.720,0.
Ponds,ell,40.,-83,500.,300000.,300000.,100.,0.
Mill_2,poi,40.,-83,5.
```

Where the header line contains file format version and the number of sources in the file and one line for each source containing all the information for each source in comma delimited format. The relevant information for each source shape is (order is important):

- Rectangular: **srcname**, **shape**, **srcy**, **srcx**, **srcz**, **xsize**, **ysize**, **zsize**, **rotation**
- Triangular: **srcname**, **shape**, **srcz**, **polylat**, **polylon**, **zsize**
- Polygonal: **srcname**, **shape**, **srcy**, **srcx**, **srcz**, **nvert**, **polylat**, **polylon**, **zsize**
- Elliptical: **srcname**, **shape**, **srcy**, **srcx**, **srcz**, **xsize**, **ysize**, **zsize**, **rotation**
- Point: **srcname**, **shape**, **srcy**, **srcx**, **srcz**

Note the extra variable **nvert** in the polygonal source that is not in the namelist. This is the number of vertices in the polygon.

$EMISSIONS Namelist

emfile character	The UNIX path and filename containing the emission database (up to 80 characters). An example file and the file format follows this table. Note that the path is optional and can be either relative or absolute (no path is equivalent to './'). If **emfile** is set to 'none' the emission database is read from the following namelist parameters. If **specfile** points to a database file, with the exception of **iemit**, **isimend** and **ienddays**, the remaining **$EMISSIONS** namelist parameters are overwritten.
iemit integer array[†]	A multi-valued parameter that determines what emissions specified by this namelist are to be used for the current ***HYPACT*** simulation. A value of 1 activates the emission.
isource integer array[†]	A multi-valued parameter that specifies the nth source from the **$SOURCES** namelist or source file.
ispecies integer array[†]	A multi-valued parameter that specifies the nth species from the **$SPECIES** namelist or species file.
irelstrt **istrtdays** integer array[†]	Multi-valued parameters which specify the time when the emissions for each source region commences. • The first, second, and third pair of digits in **irelstrt** indicate the UTC hour, minute, and second of the beginning of the emission (i.e., hhmmss UTC). • **istrtdays** is a time offset indicating an integer number of days of delay following the time of a ***RAMS*** simulation before emission from a given source begins. It allows source emissions to begin more than 24 hours after the beginning of the ***RAMS*** simulation. For example, if a ***RAMS*** simulation begins at 1100 UTC on 20 June 1993, and an emission is to begin at 0800 UTC on 22 June 1993: • **irelstrt** would be set to 080000, indicating the actual UTC time. • **istrtdays** would be set to 1, indicating that the emission begins more than 1 but less than 2 days after the start of the ***RAMS*** simulation. The earliest time specified by **irelstrt** and **istrtdays** for an emission that is activated by **iemit** defines the start time of the ***HYPACT*** simulation.
ireldur **idurdays** integer array[†]	Multi-valued parameters that specify the duration of particle and/or concentration emission from each source. They relate closely to **irelstrt** and **istrtdays**. **ireldur** uses the same syntax for the number of hours, minutes, and seconds as **irelstrt**, while **idurdays** indicates a number of days of duration. Together, **irelstrt**, **istrtdays**, **ireldur**, and **idurdays** determine the ending time for the emission of each source. A duration time of zero or anything less than the timestep length **dtpart** causes a source to act as in instantaneous or `puff' release.

isimend **ienddays** integer	Parameters that specify the ending time of the ***HYPACT*** simulation. • **isimend** is coded in hours, minutes, and seconds as described for namelist parameter **irelstrt**, and directly indicates the UTC time (i.e., hhmmss UTC). • **ienddays** is the integer number of days beyond the beginning of the ***RAMS*** simulation when the ***HYPACT*** simulation will stop. Following the example given in the description of **irelstrt** and **istrtdays** above, if the ***HYPACT*** run were to end at 1304 UTC on 22 June 1993, **isimend** would be set to 130400, indicating the UTC hours, minutes, and seconds, while **ienddays** would be set to 2, indicating that the ***HYPACT*** run would finish at least 2, but less than 3, days after the beginning of the ***RAMS*** simulation.
Type character array[‡]	A multi-valued parameter that specifies the source type. • If set to 'lag' (Lagrangian), only Lagrangian particles will be released from the source. • If set to 'eul' (Eulerian), only Eulerian concentrations will be released from the source. • If set to 'hyb' (hybrid), particles will be transformed into Eulerian terms when the hybrid criteria (**ihybmin**, **hybhoriz** and **hybvert**) set in the **$GENERAL** namelist are met. • If set to 'both' (both), both Eulerian concentrations and Lagrangian particles will be released from the source. Fields will be calculated as if there were two sources (i.e., concentrations will be effectively doubled).

scaling
real array[†]

A multi-valued parameter that specifies how the emission rate and number of particles are determined. Any two of the three namelist variables **rate**, **numparts** and **ratio** can be used to calculate the third as follows:

1. If set to 'rat' (rate), the emission rates are determined from the particle release rate (**numparts**), the size of the source (volume, area or length determined from **xsize**, **ysize**, **zsize** and **shape** from the **$SOURCES** namelist **-** noting that the size is set to 1 for a point source), the timestep (**dtpart** from the **$GENERAL** namelist), and the mass per particle ratio (**ratio**):

```
                                  mass ratio
emission rate = particle rate * ---------------
                                size * timestep
```

   The particle rate (**numparts**) has units of particles/timestep. If the mass units are ug (**units** from the **$SPECIES** namelist) the mass ratio (**ratio**) has units of ug/particle. Thus, if the source is volumetric (size has units of m^3), the emission rate has units of $(ug/s)/m^3$ (i.e., it is a specific rate that is calculated).

2. If set to 'tot' (total), **numparts** is the total number of particles released and the particle release rate is determined from the total particles, the release duration (determined from **ireldur**, and **idurdays**) and the timestep:

```
                                      timestep
particle rate = total particles * ----------------
                                  release duration
```

   The emission rates are then determined as for **scaling** set to 'rat'.

3. If set to 'spe' (specific), the particle release rates are determined as a function of the specific emission rate (**rate**), the source size, the timestep and the mass ratio:

```
                                        timestep
particle rate = emission rate * size * ----------
                                       mass ratio
```

   The particle rate has units of particles/s.

4. If set to 'abs' (absolute), the particle release rates are determined from absolute emission rate (**rate**), the timestep and the mass ratio:

```
particle rate = emission rate * timestep / mass ratio
```

   The calculate particle rate has units of particles/s.

5. If set to 'spe' (specific) and the mass ratio (**ratio**) is set to 0, the mass ratios are determined as a function of the specific emission rate (**rate**), the size of the source, the timestep and the particle release rates (**numparts**):

```
                                     timestep
mass ratio = emission rate * size * -------------
                                    particle rate
```

   If the mass units in the emission rate are defined as 'ug', the derived mass ratio has units of ug/particle.

6. If set to 'abs' (absolute) and the mass ratio (**ratio**) is set to 0, the mass ratios are determined from absolute emission rate (**rate**), the timestep and the particle release rates (**numparts**):

```
                                timestep
mass ratio = emission rate * -------------
                             particle rate
```

   If the mass units in the emission rate are defined as 'ug', the derived mass ratio has units of ug/particle.

Note that, as whole numbers of particles only can be emitted, particle emission rates from timestep to timestep are adjusted to maintain correct average emission rates. A set of emissions examples is included below.

Rate real array[†]	A multi-valued parameter that specifies the emission rates if **scaling** is equal to 'spe' or 'abs'. • If **scaling** is equal to 'spe', **rate** is the specific emissions rate. In this case, if the mass units in the emission rate are defined as 'ug', and the source is volumetric, the units of **rate** are (ug/s)/m^3. • If **scaling** is equal to 'abs', **rate** is the absolute emissions rate. In this case, if the mass units in the emission rate are defined as 'ug', the units of **rate** are ug/s. If **scaling** is equal to 'rat', the emission rate is calculated. The same mass units must be used for **rate** and **ratio**, which should also be consistent with the **$SPECIES** namelist variable, **units**.
Numparts integer array[†]	A multi-valued parameter that specifies the number of particles emitted per ***HYPACT*** timestep (**dtpart**) if **scaling** is equal to 'rat', or if **scaling** is equal to 'spe' or 'abs' and **ratio** is equal to 0. If **scaling** is equal to 'spe' or 'abs' and **ratio** has a positive value, the number of particles is calculated. Note that the total particles emitted for all sources cannot exceed the **$GENERAL** namelist variable, **maxpart**.
Ratio real array[†]	A multi-valued parameter which specifies the mass to particle ratio used to calculate the number of particles released per timestep if **scaling** is equal to 'sca', or the emission rate if **scaling** is equal to 'tot'. If **ratio** is set 0 the mass ratio is calculated. The same mass units must be used for **ratio** and **rate**, which should also be consistent with the **$SPECIES** namelist variable, **units**.

[†] Each array must contain entries for each emission.

Emissions Database File

The following illustrates some emissions in the database file format:

1,4

```
4,2,120000,0,060000,0,both,rat,20,20.,1.e9
1,2,120000,0,060000,0,lag,abs,1,.05,120.
1,2,130000,0,020000,0,lag,rat,5,1.,0.
4,1,130000,0,020000,0,lag,rat,3,20.,1.e9
```

Where the header line contains file format version and the number of emissions in the file with each subsequent line containing all the information for each emission in comma delimited format (order is important):

- **isource**, **ispecies**, **irelstrt**, **istrtdays**, **ireldur**, **idurdays**, **type**, **scaling**, **numparts**, **rate**, **ratio**

Note that the **$EMISSIONS** namelist variables **iemit**, **isimend** end **enddays** are not included in the emissions database file, and should always be included in the **$EMISSIONS** namelist. If they are not, the defaults are to emit all emissions and run the ***HYPACT*** simulation to the end of the ***RAMS*** run.

Emissions Examples

Preliminary Note

If memory and computational speed are issues you will want to pay close attention to the numbers of particle released in a simulation.

When converting Lagrangian particles to concentrations you want to ensure that the mass ratio of a particle results in a sufficiently defined concentration field.

For example, if the sensors you are comparing the simulation results with are good to 4 ug/m^3 and your grid size at the ground surface is 100 m square and 25 m deep (250,000 m^3), then you do not want your particle mass ratio to exceed:

```
                          4 ug       1
mass ratio = 250,000 m3 * ---- * ----------
                           m3    1 particle

           = 1,000,000 ug / particle = 1 kg / particle
```

This is not as an important factor is determining a Eulerian source as the particles are only used as a convenient method for statistically distributing the emissions throughout the source volume or area (noting that a source can be complex in shape and cross the boundaries of any number of grid cells).

This implies that you should have a large number of particles configured for a Eulerian source as:

1. This will give a statistically even distribution of particles through the source region (and no false peaks).
2. They are converted immediately to concentrations upon emission and do not accumulate in the model memory.

This is not as an important factor is determining a Eulerian source as the particles are only used as a convenient method for statistically distributing the emissions throughout the source volume or area (noting that a source can be complex in shape and cross the boundaries of any number of grid cells).

When computing the various quantities for each emission it is a good idea to run ***HYPACT*** in its source analysis mode (**ihyprun** = 1) as the full emission summary is computed and printed to screen.

Case Study Scenario

Six 10 by 10 m cooling tower plumes elevated 25 m from the ground emitting at a rate of 0.2 kg/s.

In the examples that follow, each emission specifies the cooling tower with different parameters known and unknown. In all cases the computed emission parameters compute to the save values for input into ***HYPACT***, as illustrated by the source summaries.

Example #1 - Rate

Typical Usage

If you ***do not know the emission rate***, as is often the case with arbitrary sources, the 'rate' scaling option is a good way to get a sensible emission with a known amount of particles. In this way you can ensure that you release enough particles per timestep so that you get good concentration definition in the region of most interest from a Lagrangian particle source.

Relevant Namelist Settings

```
dtpart   = 50 s
xsize    = 10 m
ysize    = 10 m
zsize    = 0 m
units    = ug
duration = 21600 s (6 hours as specified by ireldur,idurdays)
scaling  = rat(e)
numparts = 10 particles/timestep
rate     = 0 ug/m2/s (UNKNOWN - calculated by HYPACT)
ratio    = 1,000,000 ug/particle

emission rate = particle rate * mass ratio / (size * timestep)

        parts * ratio   = 10 parts   1,000,000 ug     1      timestep
rate = ---------------    -------- * ------------ * ------ * --------
       size * timestep    timestep     particle     100 m2     50 s

              ug
     = 2,000 ----
             m2 s

     = 200,000 ug / s = 0.2 kg / s
```

Source Summary Output

```
Model Emission Source:   1
Source:   1  ISP  Species:   1  :N:CO:B:2:N:
Type definition        -  both      rat
Source shape           -  square
Location(lat/long)     -    40.79390      -73.10170
Centre (x,y,z)         -    75573.16       33045.85       100.0000     m
Height Dimension       -    0.000000     m
Horizontal Dimensions  -    10.00000     x   10.00000     m2
Size                   -    100.0000     m2
Emission times:
  start                -    0.000000     s
  end                  -    21600.00     s
  duration             -    21600.00     s
Timestep               -    50.00000     s
Timesteps              -    432.0000
Particles to release   -    10.00000     per timestep
                       -    4320.000     total for this source
Mass per particle      -    1000000.     ug/particle
Emission rate          -    2000.000     ug/m2/s
                       -    200000.0     ug/s
                       -   1.0000000E+07 ug/timestep
                       -   4.3200000E+09 ug (total for this source)
```

Example #2 - Total

Typical Usage

If you ***do not know the emission rate***, as is often the case with arbitrary sources, the 'total' scaling option is a good way to get a sensible emission with a known total amount of particles. In this way you can ensure that you release enough particles per timestep so that you get good concentration definition in the region of most interest from a Lagrangian particle source.

Relevant Namelist Settings

```
dtpart   = 50 s
xsize    = 10 m
ysize    = 10 m
zsize    = 0 m
units    = ug
duration = 21600 s (6 hours as specified by ireldur,idurdays)
scaling  = tot(al)
numparts = 4320 particles (total released fro this emission)
rate     = 0 ug/m2/s (UNKNOWN - calculated by HYPACT)
ratio    = 1,000,000 ug/particle

particle rate = total particles * timestep / release duration

                               50 s       1            parts
particle rate = 4320 parts * -------- * -------  = 10 --------
                             timestep   21600 s       timestep

emission rate = particle rate * mass ratio / (size * timestep)

        parts * ratio  = 10 parts   1,000,000 ug    1      timestep
rate = ---------------   -------- * ------------ * ------ * --------
       size * timestep   timestep     particle     100 m2     50 s

     = 2,000 ug / m2 s  = 200,000 ug / s = 0.2 kg / s
```

Source Summary Output

```
Model Emission Source:   2
Source:   2  MTP  Species:   1  :N:CO:B:2:N:
Type definition       -  both      tot
Source shape          -  square
Location(lat/long)    -    41.07310      -71.92330
Centre (x,y,z)        -    173972.5       65746.22       25.00000     m
Height Dimension      -    0.000000     m
Horizontal Dimensions -    10.00000     x   10.00000     m2
Size                  -    100.0000     m2
Emission times:
  start               -    0.000000     s
  end                 -    21600.00     s
  duration            -    21600.00     s
Timestep              -    50.00000     s
Timesteps             -    432.0000
Particles to release  -    10.00000     per timestep
                      -    4320.000     total for this source
Mass per particle     -    1000000.     ug/particle
Emission rate         -    2000.000     ug/m2/s
                      -    200000.0     ug/s
                      -   1.0000000E+07 ug/timestep
                      -   4.3200000E+09 ug (total for this source)
```

Example #3 - Specific

Typical Usage

Used when you have a ***maximum specification for the particle mass ratio*** (see the preliminary note above), and particularly useful when you know the release rate per m^2 (or in whatever spatial units the source is described in). However, you do not have or require control on the number of particles being released.

Relevant Namelist Settings

```
dtpart   = 50 s
xsize    = 10 m
ysize    = 10 m
zsize    = 0 m
units    = ug
duration = 21600 s (6 hours as specified by ireldur, idurdays)
scaling  = spe(cific)
numparts = 0 particles/s (UNKNOWN - calculated by HYPACT)
rate     = 2,000 ug/m2/s
ratio    = 1,000,000 ug/particle

particle rate = emission rate * size * timestep / mass ratio

                2,000 ug              50 s       particle
particle rate = -------- * 100 m2 * -------- * ------------
                  m2 s              timestep   1,000,000 ug

              = 10 parts / timestep
```

Source Summary Output

```
Model Emission Source:   3
Source:   3  LGA  Species:   1  :N:CO:B:2:N:
Type definition       -  both      spe
Source shape          -  square
Location(lat/long)    -    40.77920      -73.88000
Centre (x,y,z)        -    10097.79       31032.92       25.00000     m
Height Dimension      -    0.000000     m
Horizontal Dimensions -    10.00000     x   10.00000     m2
Size                  -    100.0000     m2
Emission times:
  start               -    0.000000     s
  end                 -    21600.00     s
  duration            -    21600.00     s
Timestep              -    50.00000     s
Timesteps             -    432.0000
Particles to release  -    10.00000     per timestep
                      -    4320.000     total for this source
Mass per particle     -    1000000.     ug/particle
Emission rate         -    2000.000     ug/m2/s
                      -    200000.0     ug/s
                      -   1.0000000E+07 ug/timestep
                      -   4.3200000E+09 ug (total for this source)
```

Example #4 - Absolute

Typical Usage

Used when you have a ***maximum specification for the particle mass ratio*** (see the preliminary note above), and particularly useful with complex source shapes where you know the total release rate. However, you do not have or require control on the number of particles being released.

Relevant Namelist Settings

```
dtpart   = 50 s
xsize    = 10 m
ysize    = 10 m
zsize    = 0 m
units    = ug
duration = 21600 s (6 hours as specified by ireldur, idurdays)
scaling  = abs(olute)
numparts = 0 particles/s (UNKNOWN - calculated by HYPACT)
rate     = 200,000 ug/s
ratio    = 1,000,000 ug/particle

particle rate = emission rate * timestep / mass ratio

                200,000 ug     50 s       particle
particle rate = ---------- * -------- * ------------
                    s        timestep   1,000,000 ug

              = 10 parts / timestep
```

Source Summary Output

```
Model Emission Source:   4
Source:   4  POU  Species:   2  :N:S0:B:2:N:
Type definition       -  both      abs
Source shape          -  square
Location(lat/long)    -    41.62560      -73.88190
Centre (x,y,z)        -    9810.993       125092.3       25.00000     m
Height Dimension      -    0.000000     m
Horizontal Dimensions -    10.00000     x   10.00000     m2
Size                  -    100.0000     m2
Emission times:
  start               -    0.000000     s
  end                 -    21600.00     s
  duration            -    21600.00     s
Timestep              -    50.00000     s
Timesteps             -    432.0000
Particles to release  -    10.00000     per timestep
                      -    4320.000     total for this source
Mass per particle     -    1000000.     ug/particle
Emission              -    200000.0     ug/s
                      -    2000.000     ug/m2/s
                      -   1.0000000E+07 ug/timestep
                      -   4.3200000E+09 ug (total for this source)
```

Example #5 - Specific

Typical Usage

Used when the ***particle mass ratio is unknown*** and particularly useful when you know the release rate per m^2 (or in whatever spatial units the source is described in). You also have control on the number of particles being released per timestep.

Relevant Namelist Settings

```
dtpart   = 50 s
xsize    = 10 m
ysize    = 10 m
zsize    = 0 m
units    = ug
duration = 21600 s (6 hours as specified by ireldur, idurdays)
scaling  = spe(cific)
numparts = 10 particles/s
rate     = 2,000 ug/m2/s
ratio    = 0 ug/particle (UNKNOWN - calculated by HYPACT)

mass ratio = emission rate * size * timestep / particle rate

             2,000 ug              50 s          s
mass ratio = -------- * 100 m2 * -------- * ------------
               m2 s              timestep   10 particles

           = 1,000,000 ug / particle  = 1 kg / particle
```

Source Summary Output

```
Model Emission Source:   5
Source:   5  BLM  Species:   2  :N:S0:B:2:N:
Type definition       -  both      spe
Source shape          -  square
Location(lat/long)    -    40.18330      -74.13330
Centre (x,y,z)        -   -11317.39      -35184.75       25.00000     m
Height Dimension      -    0.000000     m
Horizontal Dimensions -    10.00000     x   10.00000     m2
Size                  -    100.0000     m2
Emission times:
  start               -    0.000000     s
  end                 -    21600.00     s
  duration            -    21600.00     s
Timestep              -    50.00000     s
Timesteps             -    432.0000
Particles to release  -    10.00000     per timestep
                      -    4320.000     total for this source
Mass per particle     -    1000000.     ug/particle
Emission rate         -    2000.000     ug/m2/s
                      -    200000.0     ug/s
                      -   1.0000000E+07 ug/timestep
                      -   4.3200000E+09 ug (total for this source)
```

Example #6 - Absolute

Typical Usage

Used when the ***particle mass ratio is unknown*** and particularly useful with complex source shapes where you know the total release rate. You also have control on the number of particles being released per timestep.

Relevant Namelist Settings

```
dtpart   = 50 s
xsize    = 10 m
ysize    = 10 m
zsize    = 0 m
units    = ug
duration = 21600 s (6 hours as specified by ireldur, idurdays)
scaling  = abs(olute)
numparts = 10 particles/s
rate     = 200,000 ug/s
ratio    = 0 ug/particle (UNKNOWN - calculated by HYPACT)

mass ratio = emission rate * timestep / particle rate

             200,000 ug     50 s       timestep
mass ratio = ---------- * -------- * ------------
                 s        timestep   10 particles

           = 1,000,000 ug / particle  = 1 kg / particle
```

Source Summary Output

```
Model Emission Source:   6
Source:   6  ACY  Species:    2  :N:S0:B:2:N:
Type definition       -  both      abs
Source shape          -  square
Location(lat/long)    -    39.46470      -74.58670
Centre (x,y,z)        -   -50337.31      -114885.1       25.00000     m
Height Dimension      -    0.000000     m
Horizontal Dimensions -    10.00000     x   10.00000     m2
Size                  -    100.0000     m2
Emission times:
  start               -    0.000000     s
  end                 -    21600.00     s
  duration            -    21600.00     s
Timestep              -    50.00000     s
Timesteps             -    432.0000
Particles to release  -    10.00000     per timestep
                      -    4320.000     total for this source
Mass per particle     -    1000000.     ug/particle
Emission              -    200000.0     ug/s
                      -    2000.000     ug/m2/s
                      -   1.0000000E+07 ug/timestep
                      -   4.3200000E+09 ug (total for this source)
```

HYPACT Output Files

Filename convention notes:

1. "*" following filenames represents the date string yyyy-mm-dd-hhmm.
2. "#" represents the grid number.
3. In the following examples the **$OUTPUT** namelist variable **hyppref** = 'a.a'.

HYPACT Specification File

FORTRAN free format ASCI file *a.spec*.

Main header line:

```
File Version ........................ i
Number of Sources ................... npsources
Number of Species ................... nspecies
Average Source Location (lat, lon) .. xavg, yavg
Average Source Height ASL ........... zavg
HYPACT Run Start .................... iy, imn, id, ih, im
```

Then for each source:

```
Source ID ........................... psource(nsrc)
Source Type ......................... type(nsrc)
Source Name ......................... srcname(nsrc)(1:indx1)
Source Location (lat, lon) .......... srcy(nsrc), srcx(nsrc)
Source Height (AGL, ASL) ............ srcz(nsrc), zasl(nsrc)
Species ID .......................... pspecies(species(nsrc))
Species Name ........................ specname(species(nsrc))
Mass Units .......................... units(species(nsrc)
```

Hybrid Grid File

FORTRAN free format ASCI file *a.hgrid*.

Main header line:

```
File Version ......................... i
Grid Dimensions (points in x, y, z) .. nih, njh, nkh
Grid Location (lat, lon) ............. pslat, pslon
```

Followed by four blocks:

```
x Grid Coordinates (m) ..... xh(i) (i = 1 to nih)
y Grid Coordinates (m) ..... yh(j) (j = 1 to njh)
Grid Heights ASL (m) ....... zh(j) (k = 1 to nkh)
Topography Heights (m) ..... topth(i,j) (i = 1 to nh1, j = 1 to nh2)
Land Percentages ........... pctlh(i,j) (i = 1 to nh1, j = 1 to nh2)
```

HYPACT Particle Files

FORTRAN formatted ASCI Files (a.p*)

Main header line (format i3,e16.8,8i8):

```
File Version ............... i
HYPACT Time (s) ............ hytime
Year ....................... iy
Month ...................... imn
Day ........................ id
Hour ....................... ih
Minute ..................... im
RAMS Grid Searched ......... ng
Number of sources .......... nsources
Total Particles ............ ntotparts
```

Followed by the source header lines (format 4i8) - note that there can be only one species per source within the code:

```
Database Source I ........... (nsrc)
HYPACT internal Species ID ... pspecies(nsrc))
Database Species ID .......... pspecies(species(nsrc))
Source Particles ............. nsrcparts(nsrc)
```

Followed by one line for each particle (format 7e16.8,i5):

```
x Location (m) .............. atp11
y Location (m) .............. atp12
z ASL (m) ................... atp13 + htopo
z AGL (m) ...................  atp13
Mass of Particle ............ atp(l,8)
ppm Mass of Particle ........ atp(l,9)
Particle Age (s) ............ hytime - catp(l,1)
Hybrid Cluster .............. latp(l,2)
```

HYPACT Concentration and Header Files

VFILE Format Files (a.a*.g# and a.a*.head). Files output are limited to the grids selected by the **$OUTPUT** namelist parameter **irgrid**.

Meteorological variables saved (if requested by **$OUTPUT** namelist parameter **imetout**):

Topography	4
Land Percentage	16
U Wind Component	1
V Wind Component	2
W Wind Component	3
Turbulent Kinetic Energy	14
Theta	19
Vertical Diffusivity	68
Horizontal Diffusivity	69

Concentration variables saved for each species (if requested by **$OUTPUT** namelist parameters ieulout and ilagout, and the **$SPECIES** namelist parameter **ioutspec**):

Lagrangian	100 + nsp
Average Lagrangian	200 + nsp
Eulerian	300 + nsp
Average Lagrangian	400 + nsp