

Release Notes (18-Jun-2025)

Very brief summary of changes in CMFGEN and associated programs compared to the older web version. Changes are mostly minor.

Makefile_definitions has been modified, and there are now two versions that can be contained in \$cmfdist — a debug and a non-debug versions.

`make clean`

will clean all the mod, exe, and lib directories, and removes all .o files

`make dbg_clean`

will clean all the mod_dbg, exe_dbg, and lib_dbg directories, and removes all .o files.

The one caveat at present is that some .o files are not deleted. This will cause problems if you continually switch back and forth while trying to do partial makes.

All .mod file (created from compilations) are now stored in the appropriate mod directory (i.e., \$cmfdist/mod of \$cmfdist/mod_dbg).

An update to the lte routines in lte_hydro was done to reduce crashes at low temperatures (new files in lte_hydro: det_ed_v2.f, genopaeta_v10.f, writedc_v3.f). Note: Another version of genopaeta_v10.f exists in another directory.

misc/create_mod_sum.f was altered to allow abundances on a log H=12 scale to be printed.

A listing of all changes can be seen (except for the Makefile and Makefile_definitions in the main directory) can be obtained by doing the following in the main directory of the new distribution:

`$cmfdist (old cmfdist) > RUB`

The RUB file will simply contain a list of the commands that are executed. Look at Diff_sum to see a list of changes,

Release Notes (17-Feb-2025)

Very brief summary of changes in CMFGEN and associated programs compared to the older web version. There were changes to ALL files.

Major Changes

Double precision variables in CMFGEN are now specified in as REAL(KIND=LDP) where the kind variable is set in cur_cmf/tools/set_kind_module.f. This allows CMFGEN to be compiled using INTEL 80 bit arithmetic which can be useful in some SN models with molecules. On INTEL machines the code will run slower when this option is on. In principal, CMFGEN could also be run using 128 precision, but this will be very slow as such an extension is usually implemented via software. For both these cases, the supplied blas and lapack routines need to be used (i.e., not those supplied by the INTEL math libraries).

An initialization issued was fixed which occasionally caused CMF_FLUX to require way too much memory.

Handling of photoionization to excited levels was changed. This affects cases where lower levels belong to the same term are not in a single level, and when the densities are low. Primarily affects nebular epoch supernovae models.

Release Notes (28-June-2023)

Brief summary of changes in CMFGEN and associated programs compared to the older web version. In this CMFGEN release, there are over 150 files that were not in the previous release.

Compilation issues

There have been some reports of compilation issues. These issues depend on the Fortran compiler, the version of the Fortran compiler, and the operating system. CMFGEN and associated routines have been successfully compiled using different versions of gfortran (including the new macbook pros with the M2 chip), pgf95, and the INTEL compiler.

Note: For `cmfgen_dev.exe` and `cmf_flux.exe` it is highly recommended that you use the parallel versions — they are much faster than the sequential version.

Intel compiler

Use (following advice from Joachim Bestenlehner who did have success):

```
F90 = ifort -qmk=sequential
FG = -xextend-source -nbs -O3 -I$(MOD_DIR) -I$(INSTALL_DIR)obs/ -I$(MKLROOT)/include/intel64/ipp64
BLAS =$(MKLROOT)/lib/intel64/libmkl_blas95_ipp64.a -L$(MKLROOT)/lib/intel64 -lpthread -lm -ldl -liomp5
LAPACK=$(MKLROOT)/lib/intel64/libmkl_lapack95_ipp64.a -L$(MKLROOT)/lib/intel64 -lpthread -lm -ldl -liomp5
```

and parallel

```
F90 = ifort -qopenmp -qmk=parallel
FG = -xextend-source -nbs -O2 -I$(MOD_DIR) -I$(INSTALL_DIR)obs/ -I$(MKLROOT)/include/intel64/ipp64
FH = -xextend-source -nbs -O2 -I$(MOD_DIR) -I$(INSTALL_DIR)obs/ -I$(MKLROOT)/include/intel64/ipp64
FD = -xextend-source -nbs -O2 -I$(MOD_DIR) -I$(INSTALL_DIR)obs/ -I$(MKLROOT)/include/intel64/ipp64
```

```
BLAS =$(MKLROOT)/lib/intel64/libmkl_blas95_ipp64.a -L$(MKLROOT)/lib/intel64 -lpthread -lm -ldl -liomp5
LAPACK=$(MKLROOT)/lib/intel64/libmkl_lapack95_ipp64.a -L$(MKLROOT)/lib/intel64 -lpthread -lm -ldl -liomp5
```

Apparently `cmf_flux` has an issue with the linux compiler, which is why the -O2 option is used.

gfortran

There are successful links in the supplied Makefile_definitions file. Some versions of gfortran implement checks on argument consistency. There are three routines where this is a problem. In one case, I pass a scalar as a vector which is computationally okay since in this case a vector length of 1 is passed. I will try to rectify these in the near future. If there is a problem, add

-fallow-argument-mismatch
to the definition of F90 in \$cmfdist/Makefile_definitions.

gfortran (Macbook Pro, M2 chip)

I did have some issues with libraries, but the following worked.

```
F90 = gfortran -fallow-argument-mismatch -L/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/lib -O -fopenmp -m64
F90nomp = gfortran -L/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/lib -O
FG = -ffixed-line-length-0 -fno-backslash -fd-lines-as-comments -I$(MOD_DIR) -J$(MOD_DIR)
FD = -g -ffixed-line-length-0 -fno-backslash -fbounds-check -fd-lines-as-comments -ffpe-trap=invalid,zero,overflow -I$(MOD_DIR) -J$(MOD_DIR)
FFREE = -fno-backslash -I$(MOD_DIR) -J$(MOD_DIR)
FFRED = -fno-backslash -g -I$(MOD_DIR) -J$(MOD_DIR)
```

I linked the supplied blas/lapack routines, but in general the routines supplied with XCODE should be linked.

pgf95

There were issues with the V18.3 of the PGF95 compiler which were solved with the V19.7. The issues were related to both compilation and running. When running with 18.2 you get an error:

```
dyld: __dyld section not supported in /Volumes/strax/cmfgn/cur_cmfgn/exe/cmfgn_dev.exe
```

If you have compilation issues, please let me know.

Running cmf_flux.exe

The newest version of [batobs.sh](#) calls [bat_ins.sh](#) which is created by [create_batobs_ins.exe](#) using a file BAT_PARAMS. With this method it is easier to run multiple flux calculations, and the formatting of the key words and their input is not critical.

Running cmfgn_dev.exe

A program [create_batch_ins.exe](#) has been created to facilitate the use of the latest atomic data. It uses the [batch.sh](#) file from a previous model and creates a file [batch_ins.sh](#) with all the atomic data links for species other than H and He. This can then be inserted (or called) from batch.sh. The F_TO_S links may need to be edited.

CMFGEN changes

This is just a brief summary of some of the changes that may have a more direct affect on users.

Automatic means of adding additional species and ionization stages.
Requires [RVTJ](#), [EDDFACTOR](#), and [EDDFACTOR_INFO](#) files from exiting model.

New routines for SN gamma-ray transport.
Method for adding shock power (SN)

ABS_MEAN and PLANCK_MEAN output to RVTJ. As a consequence the new RVTJ file cannot be read by older versions of CMFGEN. Future changes to RVTJ should not have this effect. Older versions could be updated by implementing calls to `$cmfdist/tools/rd_rvtj_v5.f`

H- ionization now referred to as H0 (not HM1) — done to facilitate adding molecules.

New routines are now utilized to compute quadrature designed to compute moments of the radiation field. This was done to overcome an accuracy issue when $R(n_{\text{max}}) \gg R^*$. Also affects [cmf_flux.exe](#). Should be transparent to users, but this was a late change to nearly a dozen routines in the current release.

CMFGEN options

More details on these options will be provided in the CMFGEN documentation.

For associated keyword, look at [\\$cmfdist/new_main/mod_sums/rd_control_variables.f](#)

[CHK_NG]	Add to the VADAT file (in NG section) and set to F(false) Still working on a robust procedure to switch this option on.
[DC_METH]	Used when reading departure coefficients from an old model. New option TR added, and is now the preferred option. Gives better estimates due to changes in T at a given depth.
[PNT_SRCE] [TEFF_PNT_SRCE]	Allow point source to be added (limited applications).
[INC_SHOCK_POWER]	For SN models (& several associated keywords)
[RD_NT-THERM_SPEC]	Reads in non-thermal electron spectrum. Set to T(true) when debugging code. As a consequence, the non-thermal spectrum will not be recomputed every time you restart the code.
[CHK_J] [CHK_H] [CHK_XM]	Options designed to improve stability of radiation transfer. Can use defaults.
[AUTO_ADD]	Allows new DC files to be created for species without XzV_IN files. Requires RVTJ , EDDFACTOR , and EDDFACTOR_INFO files from existing model, and should be used with the FIXED_J=T(true) option in VADAT. Note: Initially the changes can be huge and quite a few iterations may be needed before convergence is achieved.
[AUTO_SMOOTH]	Automatically smooths populations as we iterate (primarily of use when adding species and ionization stages). Set to T(true) for stellar work, may need to be F(false) for SN work, especially shell models. Note: You can use plt_scr.exe to fudge individual populations in a CMFGEN model (you need to halt the model, use plt_scr.exe , and then restart). This can be necessary since occasionally a few populations will go to ridiculously small values.
[USE_EHB] [COMP_EHB]	Use electron heating/cooling balance as T constraint (set to F) Compute electron heating/cooling balance as T constraint check (default is [USE_EHB])

Major CMF_FLUX changes.

To save time, the emissivity and line opacity are only computed in the first iteration.

CMF_FLUX controls (in CMF_FLUX_PARAM_INIT)

[CHK_J] [CHK_H] [CHK_XM]	Options designed to improve stability of radiation transfer. Can use defaults.
[OBS_LAM_MIN] [OBS_LAM_MAX]	Restricts range at which observed spectrum is computed (not required).
[DO_CMF_EW] [DO_ALL_SOB] [SOB_EW_LAM_BEG] [SOB_EW_LAM_END]	Using a comoving-frame calculation to compute EWs. Slow — generally set to F (default) Compute all Sobolev line EWs (default is F). Start wavelength to compute line EWs (def is 900 Ang) End wavelength to compute line EWs (def is 5.0E+04 Ang)
[WR_TRANS]	Write out TRANS_INFO (long list of lines in lambda order). Default is F.

GRAMON/PGPLT package

A lot of new options, which can be listed using the 'H' option in the plot package. Read documentation and/or [\\$cmfdist/pgplt/pgplot.f](#) to see all options. Some major changes are 'CEW' and 'FEW'. The former allows line EWs to be measured using cursors. The latter can use the bounds defined by CEW to measure EWs in a large set of model spectra (together with a box file). There is a similar (modified) Gaussian fitting option - CGF and FGF.

Options were added to the [dp_curve \(curve.f\)](#) routine to pass curve labels directly to pgplot. Call [dp_curve_lab.f](#). The calls have been implemented in some options in some routines.

Options related to line ids (RID and REW) have been improved. RID reads line IDs from a file created by [dispgen.exe](#), and is most useful for photospheric lines. REW is the EW_DATA file created by [cmf_flux.exe](#), and is most useful for emission line stars (i.e., WR stars and LBVs). Line IDs are now modified so that Sk2 becomes Si II.

Files in \$cmfdist/misc

Improved version of [create_mod_sum.exe](#).
Use to create a summary file for a set of stellar models.

Improved version of [mod_cool.exe](#)
Designed to look at and study cooling (from the electron point of view).

Improved version of [mod_prr.exe](#)
Designed to study recombination balance of an ion (and hence uses XzVPRRR).

Improved version of [plt_prr.exe](#)
Designed to plot XzVPRRR file.

Improvements to [rev_rvsig.exe](#)
Designed to update a RVSIG_COL input for a new model (e.g., a change in mass loss rate, velocity law, number of depth points etc). New velocity law options added including an option to simply copy the velocity law from the input file.

DISPGEN

Many new options added and others changed.
You always look at [\\$cmfdist/com/maingen.f](#) to see/add options.
[maingen.f](#) has a relatively simple structure.

With [dispgen.exe](#) you do not need to read in all the atomic data files. If you are only interested in structure plots, or CNO, you can simply read in RVTJ, and for example, CNO. The elements are read in atomic number order. For this option, enter RVTJ(ask), and enter "" (i.e., two double quotes) when you have read in all the needed atomic data files.