FUN3D

Fully Unstructured Navier-Stokes

Current Release: **12.4-70371**
Site Last Updated: **Wed Jun 04 16:41:00 -0400 2014**

Previous (C1: Introduction) | Up | Next (C3: Grid Generation)

2. Installation

2.1. Third-Party Libraries

Introduction

There are several libraries which we use on a routine basis for applications. Here's a high-level summary:

MPI and ParMetis are required for parallel jobs.

Metis is necessary if you wish to partition meshes with the traditional party preprocessor, which is being phased out. **Note:** the Metis library that is packaged with the ParMetis library is not compatible with FUN3D! Download the stand alone Metis library and use it if you need Metis.

KSOPT, PORT, NPSOL, SNOPT, and DOT/BIGDOT are only necessary if you are going to be doing design optimization.

SUGGAR++ and DiRTlib are only required if you need to use overset (chimera) meshes; typically overset grids are only used for certain types of moving-body problems.

The TECPLOT tecio library, though not required, allows binary TECPLOT files to be written instead of larger, slower-to-load ASCII files.

CGNS is necessary only if you are working with files written in the CGNS format.

The 6-DOF library is only required if you plan on performing 6-degree-of-freedom simulations (trajectories, etc determined by integrating equation of motion ODE's).

The sBOOM library is only required if you wish to extrapolate supersonic nearfield CFD solutions to the ground. sBOOM uses an augmented Burger's equation to predict ground-based sonic boom metrics. FUN3D and sBOOM can also be used for adjoint-based design using such objective functions.

MPI

Website: www-unix.mcs.anl.gov/mpi
Website: www.open-mpi.org

If you wish to execute the codes in the FUN3D suite across multiple processors, you will need some implementation of MPI that is compliant with the MPI 2 standard. MPI may already be installed on your machine as is the case for HPC machines, or you may have to acquire and install it yourself for your particular Fortran compiler.

When configuring FUN3D, use

```
--with-mpi=/path/to/MPI
```

where `/path/to/MPI` is the directory where your MPI installation resides.

Note: At the current time, we cannot support the pre-built **Intel® MPI Library for Linux**. The Intel® library is based on MPICH 2 from Argonne National Laboratory. We suggest downloading, building, and installing that version for support of the Intel® Fortran Compiler.

ParMetis v3.1.1

Website: www-users.cs.umn.edu/TILDAkarypis/metis/parmetis

ParMetis is a parallel implementation of Metis (see below) that is now used for all parallel FUN3D jobs.

When configuring FUN3D, use

```
--with-ParMetis=/path/to/ParMetis
```

where `/path/to/ParMetis` is the directory where your ParMetis installation resides.

Important: Do not use the sequential Metis library distributed with ParMetis! If you need to configure with Metis in addition to ParMetis (to run the sequential Party pre-/post-processor), download one of the sequential Metis packages noted below and specify it separately from your ParMetis library.

PARMETIS V3.1.1 AND NEWER MPI IMPLEMENTATIONS

More recent versions of MPI, e.g. OpenMPI, internally manage the handles (e.g., communicators) differently from previous versions of MPI. Specifically, when interfacing between Fortran and C, the more recent method uses `MPI_Comm_f2c` to convert the Fortran handle into a C handle. For example, if `comm` is a valid Fortran handle to a communicator, then `MPI_Comm_f2c` returns a valid C handle to that same communicator; if `comm = MPI_COMM_NULL` (Fortran value), then `MPI_Comm_f2c` returns a null C handle; and if comm is an invalid Fortran handle, then `MPI_Comm_f2c` returns an invalid C handle.

ParMetis, a C library with Fortran wrappers, is using the previous paradigm. Consequently, when FUN3D is compiled with a newer MPI implementation and linked with ParMetis, you will get a segmentation fault during mesh partitioning that looks like this:

```
     ... Calling ParMetis (ParMETIS_V3_PartKway) ...
forrtl: severe (174): SIGSEGV, segmentation fault occurred
```

when FUN3D attempts to call ParMetis' `kmetis` routine.

To patch ParMetis's `kmetis` routine for use with newer MPI implementations, follow the instructions in the `kmetis.patch` file that resides in FUN3D's `utils` directory or download this one.

METIS V4.0.1 (FUN3D VERSION < 12.0)

Two versions of Metis are available for use with FUN3D's party utility: Metis v4.0.1 and Metis v5.0pre2. Metis v4.0.1 is recommended for small to medium sized grids (less than 16M nodes); whereas Metis v5.0pre2 is recommended for grids larger than 16M nodes. Note: Metis v4.0.1 is internally limited to 32-bit addressing.

Website: www-users.cs.umn.edu/TILDAkarypis/metis

When configuring FUN3D, use

```
--with-metis=/path/to/metis
```

where `/path/to/metis` is the directory where your metis installation resides.

METIS V5.0PRE2 (FUN3D VERSION < 12.0)

When using metis for large grids (i.e., internal numbers approaching the 32-bit limit, and/or the 2GB memory limit), then Metis v5 is recommended. This version supports portable I/O on both 32- and 64-bit architectures, and supports the larger grids. Download this version from the same site as metis v4.0.1.

When configuring FUN3D, use

```
--with-metis=/path/to/metis
```

where /path/to/metis is the directory where your metis installation resides.

KSOPT

Contact: Greg Wrenn
Documentation: Wrenn, Gregory A., "An Indirect Method for Numerical Optimization Using the Kreisselmeier-Steinhauser Function," NASA CR 4220, March 1989.

The KSOPT library is used for multiobjective and constrained FUN3D-based design optimization. It is not required if you do not plan on using the design optimization feature of FUN3D. If you configure FUN3D to link to KSOPT, you must use the Fortran90 implementation of KSOPT, with its object files gathered to a library called `libksopt.a`. Only FUN3D versions 10.7 and later support the use of KSOPT.

When configuring FUN3D, use

```
--with-KSOPT=/path/to/ksopt
```

where `/path/to/ksopt` is the directory where your KSOPT installation resides.

PORT

Website: www.bell-labs.com/project/PORT
Website: netlib.sandia.gov/master/index.html

The PORT library is used for unconstrained FUN3D-based design optimization. It is not required if you do not plan on using the design optimization feature of FUN3D. The Netlib site listed above offers a tarball of the PORT library with a Makefile. Download the tarball from Netlib, but then replace their Makefile with this one. Note that if you install both the PORT and NPSOL (below) libraries, you may have to comment out some of the low-level BLAS routines in one of the two packages; the libraries come with duplicate versions of some of the algebra routines which the linker may complain about.

When configuring FUN3D, use

```
--with-PORT=/path/to/port
```

where `/path/to/port` is the directory where your PORT installation resides.

NPSOL

Website: www.sbsi-sol-optimize.com (Must purchase)

The NPSOL library is used for constrained FUN3D-based design optimization. It is not required if you do not plan on using the design optimization feature of FUN3D. Note that if you install both the PORT (above) and NPSOL libraries, you may have to comment out some of the low-level BLAS routines in one of the two packages; the libraries come with duplicate versions of some of the algebra routines which the linker may complain about.

When configuring FUN3D, use

```
--with-NPSOL=/path/to/npsol
```

where `/path/to/npsol` is the directory where your NPSOL installation resides.

SNOPT

Website: www.sbsi-sol-optimize.com (Must purchase)

The SNOPT library is used for constrained FUN3D-based design optimization. It is not required if you do not plan on using the design optimization feature of FUN3D.

When configuring FUN3D, use

```
--with-SNOPT=/path/to/snopt
```

where `/path/to/snopt` is the directory where your SNOPT installation resides.

DOT/BIGDOT

Website: www.vrand.com (Must purchase)

The DOT/BIGDOT library is used for unconstrained or constrained FUN3D-based design optimization. It is not required if you do not plan on using the design optimization feature of FUN3D.

When configuring FUN3D, use

```
--with-DOT=/path/to/dot
```

where `/path/to/dot` is the directory where your DOT/BIGDOT installation resides.

SUGGAR++-1.0.10 OR HIGHER

Contact: Ralph Noack

NOTE: We now **require** SUGGAR++ for overset applications; FUN3D is no longer compatible with the older SUGGAR code. SUGGAR++ is faster and easier to use. Furthermore SUGGAR++, like FUN3D, can utilize FIELDVIEW and AFLR3 grid files in addition to VGRID files. These additional grid formats allow for "mixed-element" grid and 2D simulations within FUN3D. VGRID grids are limited to pure tetrahedral meshes, and precludes true 2D simulations.

SUGGAR++ is used to assemble composite meshes, cut holes, determine interpolation coefficients, etc. It is not required unless you plan to use overset (chimera) meshes in FUN3D. SUGGAR++ may be compiled as a stand-alone executable and/or as a library. For static overset meshes you will need the stand-alone compilation; for moving body simulations using FUN3D Version 10.5 and above, you will need to compile both the stand-alone executable and the library.

It is beyond the scope of this web page to provide details of compilation steps of third-party software; see the documentation that comes with SUGGAR++ for more information. However, Ralph has provided us with a couple of scripts that make compilation of the complete DiRTlib/SUGGAR++ suite very easy. You may download these scripts to compile both DiRTlib and SUGGAR++. Similar scripts for the older SUGGAR code can be found here

When configuring FUN3D, use

```
--with-suggar=/path/to/SUGGAR++
```

where `/path/to/SUGGAR++` is the directory where SUGGAR++ library archive files (`.a` files) reside. In this directory, there must be an archive file (or a link to an archive file) called `libsuggar.a`, which is the serial compilation of SUGGAR++, and there must also be an archive file (or a link to an archive file) called `libsuggar_mpi.a`, which is the mpi compilation of SUGGAR++. The above-mentioned scripts do this automatically.

DiRTlib v1.40 OR HIGHER

Contact: Ralph Noack

The DiRTlib library must be linked to FUN3D in order to utilize the overset connectivity data computed by SUGGAR++ It is not required unless you plan to use overset (chimera) meshes in FUN3D. However, if you do link DiRTlib to FUN3D you must also link SUGGAR++ to FUN3D.

It is beyond the scope of this web page to provide details of compilation steps of third-party software; see the documentation that comes with DiRTlib for

more information. However, Ralph has provided us with a couple of scripts that make compilation of the complete DiRTlib/SUGGAR++ suite very easy. You may download these scripts to compile both DiRTlib and SUGGAR++. Similar scripts for the older SUGGAR code can be found here

When configuring FUN3D, use

```
--with-dirtlib=/path/to/DiRTlib
```

where `/path/to/DiRTlib` is the directory where DiRTlib library archive files (.a files) reside. In this directory, there must be an archive file (or a link to an archive file) called `libdirt.a`, which is the serial compilation of DiRTlib, and there must also be an archive file (or a link to an archive file) called `libdirt_mpich.a`, which is the mpi compilation of DiRTlib. The above-mentioned scripts do this automatically.

TECPLOT

Website: www.tecplot.com

By default, any TECPLOT output generated from the postprocessor code party, or from within the flow solver itself, is written as an ASCII file. If you have a copy of TECPLOT, you were provided with a library archive `tecio.a` (or `tecio64.a` for 64-bit versions). Provided you have the `tecio.a` archive, you may configure the FUN3D suite to use that library via:

```
--with-tecio=/path/to/tecio
```

Then, TECPLOT solution data written out from the flow solver (FUN3D Version 10.7 and higher) and volumetric TECPLOT data written by party will be in binary form. This results in smaller file sizes and faster importation into TECPLOT.

Note, however, that when you link against a particular tecio library, you will have to use the corresponding TECPLOT visualizer. For example, if you link against the tecio library from TECPLOT360-2008, you will not be able to use the resulting `.plt` files in TECPLOT360, only with TECPLOT360-2008.

Also note: the tecio library that was shipped with TECPLOT360-2008 had a bug that will result in error messages when the binary files are written. You must get the updated library from TECPLOT's website .

CGNS

Website: www.cgns.org

The CGNS library is used for working with files written in CGNS format. (CGNS is a convention for writing machine-independent, self-descriptive data files for CFD, and includes implementation software.) FUN3D currently has some limited capability for reading and writing CGNS files. You only need the CGNS library if you are planning to take advantage of this feature. Version 2.5

of the CGNS library or greater is required for adding CGNS capability to FUN3D.

When configuring FUN3D, use

```
--with-CGNS=/path/to/cgns
```

where `/path/to/cgns` is the directory where your CGNS installation resides.

6-DOF

Contact: Nathan Prewitt

Contact Nathan Prewitt for the 6-DOF libraries used by FUN3D.

When configuring FUN3D, use

```
--with-sixdof=/path/to/sixdof
```

where `/path/to/sixdof` is the directory where your 6-DOF installation resides.

sBOOM

Contact: Sriram Rallabhandi

Contact Sriram Rallabhandi for the sBOOM library used by FUN3D.

When configuring FUN3D, use

```
--with-SBOOM=/path/to/sBOOM
```

where `/path/to/sBOOM` is the directory where your sBOOM library resides.

2.2. Compiling

The FUN3D suite of codes is configured and built via the GNU build system . Provided you have a Fortran95 compiler that supports the Fortran2003 stream I/O feature, a bare bones installation process would be,

```
tar zxf fun3d-12.2-63280.tar.gz
cd fun3d-12.2-63280
./configure
make
make install
```

To add third-party library capabilities, you must specify where those libraries are located by passing command line options of the form `--with-`

`LIB=/path/to/LIB` to `configure` where `LIB` is the name of a third party library.

For example, a parallel-processing version of FUN3D might be created with,