Compiling TAU with PETSc (Jan 2009)

TAU uses the PETSc sparse linear solver library for some modes of the linearized solvers (adjoint, frequency domain, error estimation, etc.). Some functionality of these solvers is only available with PETSc. This document describes first PETSc compilation 1)-5), and then modifications to the TAU compilation process required when linking with PETSc.

1) Download PETSc from homepage: `http://www.mcs.anl.gov/petsc/`.
   Note: The PETSc library interface sometimes changes in small ways between versions (even minor versions). Hence TAU may not compile with certain versions. At time of writing TAU SVN revision 11180 (Jan 2009) was compiling with PETSc v2.3.3-p15. This is the PETSc version recommended at time of writing (not PETSc v3.0.0!). TAU releases prior to 2008.1.1 inclusive do not compile with PETSc.

2) Unpack the tar file, creating your PETSc directory `$PETSC_DIR`.

3) PETSc uses a Python configure system started with:
   ```
   ./config/configure.py
   ```
   Use the switch `--help` to get a list of all options. The following are important:

   MPI directory:
   Always use existing MPI installation, MPICH and OpenMPI have been tried and work. Specify directory over `--with-mpi-dir=/opt/mpich` for example.

   BLAS and LAPACK:
   These are libraries of linear algebra routines for small dense matrices written in FORTRAN. I have never successfully compiled PETSc with my own, or an existing installation of BLAS/LAPACK. Many problems come from the lack of a standardized LAPACK installation. The most robust way is to let PETSc download these libraries itself, with the switches:
   ```
   --download-f-blas-lapack=yes
   --download-c-blas-lapack=yes
   ```
   which download and compile the FORTRAN and C (over f2c) versions of the libraries respectively. With the C version no FORTRAN compiler is required, but LAPACK will not support complex-valued matrices (not necessary for almost all TAU functionality). Libraries will be placed in `$PETSC_DIR/externalpackages`, and will not be installed with the PETSc "make install" command.

   X11:
   Switch off dependency on X11 libs with `--with-x=0`. No important PETSc functionality is lost.

   FORTRAN:
   If PETSc can't automatically find your FORTRAN installation, you will likely not be successful in trying to tell it where it is. Better switch to `--download-c-blas-lapack=yes`.

   Install directory:
   Use `--prefix=<dir>` to specify install dir as usual.

E.g. a complete configure line might look like:

```
./config/configure.py --with-mpi-dir=/opt/mpich \
--download-f-blas-lapack=yes --with-x=0 --prefix=/home/han/path
```

Configure will take a long time as it also has to compile LAPACK. When finished it will ask you to set 2 environment variables, `$PETSC_ARCH` and `$PETSC_DIR`, do this in the current shell with the values configure gives you. E.g. for bash:

```
PETSC_ARCH=linux-gnu-c-debug; export PETSC_ARCH
PETSC_DIR=<wherever>; export PETSC_DIR
```

Note that these variables are only necessary for PETSc compilation, not for linking later. It is not necessary to add them to your standard environment (though it may be useful).

4) Compile PETSc with: `make all test`.

5) Installation. First install basics with `make install`. This copies only PETSc libraries and headers to your `--prefix` path, not BLAS/LAPACK and not PETSc architecture-specific headers, so:

   a) Copy the BLAS and LAPACK libs into your `--prefix` library path (together with the PETSc libs). They are found in:
   `$PETSC_DIR/externalpackages/fblaslapack/$PETSC_ARCH/libblas.a`
   If you're using the C version they will have slightly different names.

   b) Copy `petscconf.h` from `$PETSC_DIR/bmake/$PETSC_ARCH` into your `--prefix` install path (together with other PETSc headers).

   If you are intending to use multiple builds of PETSc for different architectures, then you will likely want to keep these architecture-specific files separate.

6) TAU configuration. Add into your `.taudef` file the following lines (without line breaks):

```
PETSC_DEF = -DHAVE_PETSC -DHAVE_LAPACK -DHAVE_F_LAPACK
PETSC_INC = -I<prefix>/include
PETSC_LIB = -L<prefix>/lib/<arch>
PETSC_LIB_NAME = -lpetsccontrib -lpetscsnes -lpetscksp -lpetscdm -
lpetscmat -lpetscvec -lpetsc

BLAS_LIB = -L<prefix>/lib/<arch>
LAPACK_LIB = -L<prefix>/lib/<arch>
BLAS_LIB_NAME = -lblas -lgfortan
LAPACK_LIB_NAME = -llapack
```

Where `<prefix>` is your `--prefix` path and `<arch>` is your `$PETSC_ARCH`. Note that the names of the BLAS/LAPACK libraries may be different than given here, check what they are in your case.

If you have the C version of BLAS/LAPACK, then remove `-DHAVE_F_LAPACK` from the `PETSC_DEF` variable. Otherwise you will get error messages when linking the TAU solver about missing routines `zgeev_()` etc. which are only available in the FORTRAN version of LAPACK.

Depending on your FORTRAN installation the `-lgfortran`, entry must be modified. With older versions of the GNU FORTRAN compiler it might be `-lg2c`. For other

installations, you may also need to specify the directory, e.g. for the Intel FORTRAN compiler on my PC I set:

```
BLAS_LIB_NAME = -lblas -L/opt/intel/fce/9.1.039/lib -lifcore
```

7) TAU compilation. Compile as usual. PETSc is only used in the solver module, so if:

```
make taulib euler
```

is successful you have successfully compiled TAU with PETSc!