Instructions for running Implicit KEM version of XGC-1 example on NERSC

Directory:

The necessary files are currently located in:

```
/global/homes/b/bsturd/xgc_kem
```

All NERSC users should be able to access this directory and copy it to their scratch directory.

There are three subdirectories in `xgc_kem`:

```
ml_data   run    src
```

`ml_data` contains the data necessary for the ML approaches in section 7 of the write up. Some manipulation of the data contained in these file will be necessary to construct error/residual pairs.

`run` contains the input files, etc. needed to run the executable `xgc-kem`

`src` contains the source code along with a makefile

Building:

The Makefile included in the src directory can be used to build the KEM version of XGC-1 on Cori or Edison. Currently the necessary libraries are located in the following directories:

```
/project/projectdirs/m499/rhager/software/cori_haswell/intel

/project/projectdirs/m499/bsturd/software
```

I am happy to work to install these elsewhere for users without access to the m499 project directory.

You will need the following modifications to your environment (can be included in .bashrc.ext):

```
module unload cray-libsci     # to avoid conflicts with mkl
module load cray-hdf5-parallel
```

(For Matlab diagnostics on Cori)

```
export MATLABPATH=/global/homes/b/bsturd/software/matlab/cori     #
for Matlab diagnostics
```

```
module load matlab
```

(For Matlab diagnostics on Edison)

```
export MATLABPATH=/global/homes/b/bsturd/software/matlab/edison     #
for Matlab diagnostics

module load matlab
```

Running:

After building, you will need to place a copy of the executable `xgc-kem` into the run directory. All input files needed for the test case can be found in the run directory. There is included a batch script called "job", which is currently set up to run on Cori (Haswell nodes). If you are not using the m499 account, you will need to modify the following line:

```
#SBATCH -A (your account number)
```

To run on Edison, simply remove or comment out the line

```
#SBATCH -C haswell
```

To run, simply enter:

```
sbatch job
```

Diagnostics:

After running `xgc-kem`, a number of output files will be written to the `run` directory. Matlab diagnostic scripts contained in the `diag` directory are given to read in files from the `run` directory and generate plots. When we set `diag_3d_precond_iter=.t.` in the input file, a diagnostic file in .bp format will be written at each iteration with naming convention `xgc.3d.xxxxx.bp`. Here, the file number `xxxxx` contains the time step and iteration number. For example, suppose the file is `xgc.3d.00312.bp`. Then the file contains the results from iteration `12` when solving for the solution at time step `3`. These files can be viewed using the Matlab function `x2_bp` using:

```
x2_bp(fnum,pnum,diagnostic)
```

Here, `fnum` is the file number (e.g. 00312), `pnum` is the poloidal plane number, and `diagnostic` is the name of the quantity you wish to view. Some options for `diagnostic` include:

- apar_try : $A_{\parallel}^{k}$
- apar_res : R_{A}^{k}

- apar_del : $(P^{-1}r^k)_{A_{||}}$
- pot_try : ϕ^k
- pot_res : R_ϕ^k
- pot_del : $(P^{-1}r^k)_\phi$

where the file number corresponds to iteration $k+1$. Recall that the iteration scheme is:

$$\phi^{k+1} = \phi^k + (P^{-1}r^k)_\phi$$
$$A_\parallel^{k+1} = A_\parallel^k + (P^{-1}r^k)_{A_{||}}.$$

A complete list of variables can be obtained by running

`bpls xgc.3d.xxxxx.bp` (requires adios module).

In addition to the .bp output files, there is a file named `res_norm.txt`, which contains the 2-norm of the residual as a function of the iteration number. This data can be plotted using the Matlab script `residual_plot.m`

Finally, a useful diagnostic is to compare the iteration data with a converged solution. For this purpose, converged solutions for the first five time steps are included in the files `run/xgc.3d.0000x.c.bp`,

where `x` is the time step number.

The Matlab scripts `pc_diag_apar.m` and `pc_diag_phi.m` produce plots of the true error at the current iteration (difference between values at the current iteration and converged values), update at the current iteration $P^{-1}r^k$, and the residual r^k for $A_\parallel$ and ϕ, respectively. For a "good" preconditioner, $P^{-1}r^k$ should closely approximate the true error at the current iteration. To select the iteration number, adjust the value of

```
pfile = '../xgc.3d.xxxxx.bp';
```

and to select the poloidal plane, adjust

```
pnum = 'x';
```

in the Matlab scripts.