Getting Started with PETSc for PDEs

configuring and running your first code

Ed Bueler, UAF

19 January 2016

seminar organization

- because you showed up today, you get a free book
  - the first five chapters are done-ish
  - I will have more chapters for you by March-ish
- you don't have to take this for credit
  - if you do want 1.0 Math 692 credit (pass/fail) then plan to write at least one *working* example PDE code using PETSc
    - easy because I provide lots of examples
    - any language you want (C, FORTRAN, python, . . .), if it has PETSc bindings

book and codes are online

- the LaTeX sources and C example codes from my book are at
  - `github.com/bueler/p4pdes`
- find a PDF for the book at the `book.pdf` link on the latest-release page
  - `github.com/bueler/p4pdes/releases`
- this seminar is a blog:
  - `bueler.github.io/p4pdes`
- these LaTeX Beamer PDF slides are generated from the blog (markdown page) by

```
pandoc -t beamer blogpage.md -o slides.pdf
```

  - no `.tex` file required
  - cool, huh?

one example from the book

- in Chapter 5 of the book I look at the nonlinear p-Laplacian PDE

$$-\nabla \cdot \left(|\nabla u|^{p-2} \nabla u \right) = f$$

on a square

$$\Omega = (0,1) \times (0,1)$$

- the PDE is approximated by a finite element on a grid, which we can distribute in parallel across processors
- the program is in C, about 300 lines, and includes just one C library

```
#include <petsc.h>
```

so: an action-filled demo of PETSc

- before helping *you* install PETSc and run examples, I will show you what it looks like to
  - compile,
  - run in serial,
  - run in parallel,
  - ask for help with,
  - visualize the output of,
  - and evaluate performance of

this PETSc code `c/ch5/plap.c`

- now, and at any time today: PLEASE ask questions at any time!

download PETSc

- PETSc download page
  - `www.mcs.anl.gov/petsc/download/index.html`
- either use Git to download:

```
$ git clone https://bitbucket.org/petsc/petsc.git
$ cd petsc/
```

- or get the "lite" version as `tar.gz`, and unpack it

```
$ tar -xzvf petsc-lite-3.6.3.tar.gz
$ cd petsc-3.6.3/
```

configure PETSc

- try to configure:

```
$ export PETSC_DIR=`pwd`
$ export PETSC_ARCH=linux-c-dbg
$ ./configure --download--mpich \
  --download-triangle --with-debugging=1
```

- I'll show this right now
- actually *read* the error messages if you get them

make and test PETSc

- make and test:

```
$ make all
$ make test
```

- success if you see

```
... /ex19 run successfully with 2 MPI processes
```

- to use this PETSc installation you'll need to add these lines (or similar) to `.bashrc` (or similar):

```
export PETSC_DIR=/home/bueler/petsc-3.6.3/
export PETSC_ARCH=linux-c-dbg
```

streams benchmark

- it is a good idea to be skeptical of multi-core architectures as parallel machines
- following the PETSc advice:

```
make streams
```

PETSc source code contains examples

- look in `examples/tutorials/` source (i.e. inside `src/`) directories:

```
$ cd src/snes/examples/tutorials/
$ gedit ex5.c &
```

- but these examples are
  - often poorly well-documented
    - especially missing links to mathematical ideas, source literature, and algorithmic possibilities
  - often cluttered with whatever the last PETSc dev who touched it was interested in
  - i.e. they are exactly what the PETSc devs play with

PETSc source code contains better examples

- look in `tutorials/` source directories:

```
$ cd $PETSC_DIR/tutorials/
$ firefox HandsOnExercise.html
```

  - these are better-supported examples and I'll come back to them
- my goal for my book is to have examples which
  - are easier to use
  - are smaller and non-evolving
  - have a clearer, and explained/cited, mathematical context
- there is even some support among PETSc devs for such a book!

a first PETSc program

- see Chapter 1 in the book

```
$ cd c/ch1/
$ gedit e.c &
$ make e
$ ./e
$ ./e -help | less
$ mpiexec -n 4 ./e
$ mpiexec -n 20 ./e
```

- now I am going to talk through the code, which is only 29 lines