

PETSc

Portable, Extensible Toolkit for Scientific Computation

Karl Rupp

`me@karlrupp.net`

Freelance Computational Scientist



Workshop on Modern Scientific Computing
DTU Compute, Technical University of Denmark

March 2-3, 2016



Step 0: Install PETSc

Download and Install PETSc

```
$> git clone --depth=1 -b maint \  
      https://bitbucket.org/petsc/petsc petsc  
  
$> cd petsc/  
$> ./configure --with-cc=gcc --with-cxx=g++ \  
      --with-fc=gfortran --download-fblaslapack \  
      --download-mpich  
$> make PETSC_DIR=... PETSC_ARCH=...  
$> make test PETSC_DIR=... PETSC_ARCH=...  
  
$> export PETSC_DIR=... PETSC_ARCH=...
```

Step 1: Hello World

Initialize Environment

```
$> export PETSC_DIR=/path/to/petsc
$> git clone https://github.com/karlsruhp/dtu2016_1.git

$> cd dtu2016_1
$> make hello

$> ./hello
$> $PETSC_DIR/$PETSC_ARCH/bin/mpirun -n 2 ./hello
```

Options

Provided via `petscrc`

Step 2: Debugging a PETSc Program

Run a Faulty PETSc Program

```
$> make debug  
$> ./debug
```

Takeaway Message

Always use `CHKERRQ(...)`

Use `gdb`

Use `valgrind`

Step 3: Vector Operations

Vector Operations

```
$> make vector  
$> $PETSC_DIR/$PETSC_ARCH/bin/mpirun -n 3 ./vector
```

Sample Output

```
Norm of y: 4.89898  
Vec Object: 3 MPI processes  
  type: mpi  
Process [0]  
4  
Process [1]  
4  
4  
Process [2]  
4  
3  
2
```

Quiz Preparation

Right answer wins chocolate!



Task

Modify program such that global vector has entries 1, 2, 3, ...

Requirement: Works independent of the number of MPI ranks

Hint: There are multiple ways to achieve this

Step 4: Poisson Equation in 2D

Poisson Equation

$$-\Delta u = f$$

```
$> make poisson2d  
$> ./poisson2d
```


Preparation

Use the following options:

```
-m 40  
-n 40
```



Minimize Iteration Count for 2 MPI Ranks

View available options with command line option `-help`

Preconditioner type specified via `-pc_type`

Do not modify KSP tolerances!

Step 5: Poisson Equation in 3D

Poisson Equation

$$-\Delta u = f$$

```
$> cd /path/to/dtu2016_1
$> git checkout reference_solutions
$> cp reference/poisson3d.c .
$> make poisson3d
$> ./poisson3d
```

Adjust Grid Size

Use the following options:

```
-da_grid_x 20  
-da_grid_y 20  
-da_grid_z 20
```

Play with Solvers

View available options with command line option `-help`

Profiling with `-log_view`

Multigrid preconditioner specified via `-pc_type mg`

Check scalability as you increase resolution in x, y, z