PyTrilinos: Epetra Maps & Vectors

Epetra Maps

- A "map" describes the distribution of global node (or element) indices across all processors
  - A "node" is a calculation point, e.g. finite elements, finite differences, etc.
  - "global" refers to a node numbering scheme that describes the entire model, i.e. what it would be if it where only run on a single processor
  - "local" refers to a the on processor node numbering scheme
- 4 ways to construct a `map`
  - `Map(numGE, iBase, comm)`
  - `Map(numGE, numME, iBase, comm)`
  - `Map(numGE, myGEs, iBase, comm)`
  - `Map(map)`

Epetra.Map() Example 1

EpetraMap1.py

```
#!/usr/bin/env python
from PyTrilinos import Epetra
comm = Epetra.PyComm()

stdMap = Epetra.Map(10, 0, comm)
print ("My global indices are: "
       + str(stdMap.MyGlobalElements()))
```

```
>>mpiexec -np 2 ./EpetraMap1.py
My global indices are:  [0 1 2 3 4]
My global indices are:  [5 6 7 8 9]
```

Epetra.Map() Example 2

EpetraMap2.py

```
#!/usr/bin/env python
from PyTrilinos import Epetra
comm = Epetra.PyComm()
number_of_global_elements = 20
number_of_elements_on_first_processor = 4

if comm.MyPID() == 0:
    number_of_my_elements = number_of_elements_on_first_processor
else:
    number_of_my_elements = (number_of_global_elements -
            number_of_elements_on_first_processor) / (comm.NumProc() - 1)

stdMap = Epetra.Map(number_of_global_elements,
        number_of_my_elements, 0, comm)
print ("My global indices are: "
        + str(stdMap.MyGlobalElements()))
```

```
>>mpiexec -np 2 ./EpetraMap2.py
My global indices are:  [0 1 2 3]
My global indices are:  [ 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]


>>mpiexec -np 3 ./EpetraMap2.py
My global indices are:  [0 1 2 3]
My global indices are:  [ 4 5 6 7 8 9 10 11]
My global indices are:  [12 13 14 15 16 17 18 19]
```

Epetra.Map() Example 3

EpetraMap3.py

```
#!/usr/bin/env python
from PyTrilinos import Epetra
comm = Epetra.PyComm()

if comm.MyPID() == 0:
    my_global_elements = [0,2,4]
else:
    my_global_elements = [1,3,5,6]

stdMap = Epetra.Map(7, my_global_elements, 0, comm)
print ("My global indices are: "
       + str(stdMap.MyGlobalElements()))
```

```
>>mpiexec -np 2 ./EpetraMap3.py
My global indices are:  [0 2 4]
My global indices are:  [1 3 5 6]
```

Epetra Vectors

- Inherits from Numpy's `numpy.ndarray`
  - Epetra Vectors *are* Numpy arrays
- Distributed over processors according to the associated `Epetra.Map()`
- Type: `double`
  - See `Epetra.IntVector()` for `int` type
- Several useful constructors
  - `Vector(map, zeroOut=True)`
  - `Vector(map, array)`
  - `Vector(vector)`

Epetra.Vector() Example 1

EpetraVector1.py

```
#!/usr/bin/env python
from PyTrilinos import Epetra
import numpy as np
comm = Epetra.PyComm()

stdMap = Epetra.Map(10, 0, comm)

x = Epetra.Vector(stdMap)

x[:] = np.arange(stdMap.NumMyElements())

print("My (local, global) indices are: "
        + str(zip(x, stdMap.MyGlobalElements())))
```

```
>>mpiexec -np 2 ./EpetraVector1.py
My (local, global) indices are:  [(0.0, 0), (1.0, 1), (2.0, 2), (3.0,
3), (4.0, 4)]
My (local, global) indices are:  [(0.0, 5), (1.0, 6), (2.0, 7), (3.0,
8), (4.0, 9)]
```

Epetra.Vector() Example 2

EpetraVector2.py

```
#!/usr/bin/env python
from PyTrilinos import Epetra
import numpy as np
comm = Epetra.PyComm()
stdMap = Epetra.Map(10, 0, comm)

x = Epetra.Vector(stdMap)
y = Epetra.Vector(stdMap)

x[:] = np.arange(stdMap.NumMyElements())
y[:] = x + 1

tmp1 = x.Dot(y)
tmp2 = x.Norm2()
tmp3 = y.MaxValue()

if comm.MyPID() == 0:
    print("x . y = " + str(tmp1))
    print("The L2 norm of x is = " + str(tmp2))
    print("The max value of y is = " + str(tmp3))
```

```
>>mpiexec -np 2 ./EpetraVector2.py
x .  y = 80.0
The L2 norm of x is = 7.74596669241
The max value of y is = 5.0
```

Epetra Comm MPI-style methods

- Several MPI-style methods available
  - `Broadcast(numpy.ndarray, int root)`
  - `GatherAll(PyObject obj) -> numpy.ndarray`
  - `SumAll(PyObject obj) -> numpy.ndarray`
  - `MaxAll(PyObject obj) -> numpy.ndarray`
  - `MinAll(PyObject obj) -> numpy.ndarray`
- All MPI methods available via
  - `GetMpiComm`
    - Example: `comm.GetMpiComm.Scatter(PyObject obj, root=0)`

Epetra.Comm() Example

EpetraCommGather.py

```
#!/usr/bin/env python
from PyTrilinos import Epetra
import numpy as np
comm = Epetra.PyComm()


x = np.arange(5)

tmp1 = comm.GatherAll(x)

if comm.MyPID() == 0:
    print(tmp1)
```

Epetra BlockMap and Epetra MultiVector

- Epetra Maps *are* Epetra BlockMaps
  - With Point Size = 1
  - Most `Map()` methods are defined for `BlockMaps()`
- Epetra Vectors *are* Epetra MultiVectors
  - With number of dimensions = 1
  - Most `Vector()` methods are defined for `MultiVectors()`