

# PyTrilinos: Epetra Matrices, Graphs, & Extensions

ME 4953/5013 - Introduction to High-Performance Computing



- `Epetra.SerialDenseMatrix`
  - Essentially an Epetra interface to BLAS
- `Epetra.CrsMatrix`
  - Column-row spase data structure
  - See also: `FE_CrsMatrix`
- `Epetra.VbrMatrix`
  - Variable-block row sparse data structure
  - See also: `FE_VbrMatrix`

EpetraSerialDenseMatrix.py

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

mat1 = Epetra.SerialDenseMatrix(3,3)
mat1.Random()
mat2 = Epetra.SerialDenseMatrix([[1,2,3],
    [4,5,6],[7,8,9]])
mat3 = Epetra.SerialDenseMatrix(mat1)
#mat3 = mat1.mat2
mat1.Apply(mat2,mat3)

print mat3
```

```
>> ./EpetraSerialDenseMatrix.py
[[-0.73692442 0.06553447 0.35772943]
 [0.51121064 -0.56208163 0.35859281]
 [-0.08269974 -0.90591077 0.86938579]]
```

EpetraCrsMatrix.py

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

numRows = 9
stdMap = Epetra.Map(numRows, 0, comm)
A = Epetra.CrsMatrix(Epetra.Copy, stdMap, 3)
for gid in stdMap.MyGlobalElements():
    if gid in (0,numRows-1):
        A.InsertGlobalValues(gid, [1], [gid])
    else:
        A.InsertGlobalValues(gid, [-1,2,-1], [gid-1,gid+1])

A.FillComplete()

x = Epetra.Vector(stdMap)
b = Epetra.Vector(stdMap)
b.PutScalar(1.0)

A.Multiply(False, b, x)

print x
```

```
>> mpiexec -np 2 ./EpetraCrsMatrix.py
[1 0 0 0 0 0 0 0 1]
```

- In the setting of matrices, graphs are mathematical constructs that describe the “connectivity” or sparsity pattern of the matrix
- `Epetra.CrsGraph`
  - Can be constructed in much the same way as `Epetra.CrsMatrix`
  - Can be used to construct a `Epetra.CrsMatrix`
  - Can be accessed from a `Epetra.CrsMatrix`, `A`, with `A.Graph()`

- Graph coloring algorithms
- Matrix-Matrix functions
  - Add
  - Multiply
- Parallel file I/O
  - Read from: Matlab, MatrixMarket file formats
  - Write to: Matlab, MatrixMarket
  - Other: HDF5, XML

```
EpetraExtExample.py
```

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

numRows = 9
stdMap = Epetra.Map(numRows, 0, comm)
A = Epetra.CrsMatrix(Epetra.Copy, stdMap, 3)
for gid in stdMap.MyGlobalElements():
    if gid in (0,numRows-1):
        A.InsertGlobalValues(gid,[1],[gid])
    else:
        A.InsertGlobalValues(gid,[-1,2,-1],[gid-1,gid,gid+1])

A.FillComplete()

graphA = A.Graph()

B = Epetra.CrsMatrix(Epetra.Copy, graphA)
C = Epetra.CrsMatrix(Epetra.Copy, graphA)
B.PutScalar(2.0)

EpetraExt.Multiply(A, False, B, False, C)
EpetraExt.RowMatrixToMatlabFile("matrix.mat", C)
```