PyTrilinos: AztecOO

ME 4953/5013 - Introduction to High-Performance Computing

- Library of algorithms for the iterative solution of large sparse systems
  - Solvers: CG, GCS, BiCGSTAB, GMRES, TFQMR
  - Preconditioners: point Jacobi, block Jacobi, Gauss-Seidel
- Supports CSR and VBR sparse matrix types

- Object-oriented interface to Aztec
- Provides a convenient interface to Epetra through `Epetra.LinearSystem()`
- Basic Use:
  - Construct coefficient matrix `A` as `Epetra.CsrMatrix` or `Epetra.VbrMatrix`
  - Construct right-hand side vector `b` as `Epetra.Vector`
  - Construct solution vector `x` as `Epetra.Vector`
  - Construct `Epetra.LinearProblem(A, x, b)`
  - Instantiate AztecOO solver
  - Pass in AztecOO parameters as `Teuchos.ParameterList`
  - Solve

AztecOO Example

1DLaplace.py

```
#!/usr/bin/env python
from PyTrilinos import Epetra
from PyTrilinos import AztecOO
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()

x = Epetra.Vector(stdMap)
b = Epetra.Vector(stdMap)
#Boundary conditions
if comm.MyPID() == 0:
    b[0] = -1
if comm.MyPID() == comm.NumProc()-1:
    b[-1] = 1

linearProblem = Epetra.LinearProblem(A, x, b)
solver = AztecOO.AztecOO(linearProblem)
solver.Iterate(10000,1.e-5)

print x
```