Interfacing PETSc to NIMROD for Linear Solver Enhancements

S. Vadlamani-Tech-X

Acknowledgments:
S.Kruger-Tech-X
C. Sovinec- Univ.of Wisconsin
B. Smith-Argonne (PETSc Team)
S. Balay-Argonne (PETSc Team)

Linear solvers for NIMROD major bottle neck for scaling

- **Direct sparse solvers have been effective for poloidal preconditioning**
  - handles the higher conditioned matrices
- **Limited parallel scaling for large polynomial degree cases has been shown, ie. ELM cases**
  - associated with NIMROD allreduces (*addressed*)
  - associated with SuperLU communcation routines
- **Memory foot-print can be restricting**
  - First solution was use of SuperLU distributed interface
  - C. Sovinec recently implemented more point-to-point communications alleviating most allreduces

PETSc is a natural choice for numerical packages

- **Modular concept for algebraic solves**
  - distributed objects
    - vector objects ---> right hand side
      - no memory copy
    - matrix objects
      - caches elements so memory copy into PETSc format
    - KSP object
      - creation of Krylov-space solver
      - including matrices used for preconditioning matrices
- **A very large suite of PETSc internal diagnostics**
  - visual, std out, and file based
  - monitor norms of residuals, matrix structure
- **Command line options**
  - mpirun -n 2 nimrod -ksp_type preonly -mat_type superlu_dist -pc_type lu

PETSc is interfaced to PETSc for real matrices

- **All wrapper subroutines in *externals/petsc.F***
  - minimize necessary preprocessing for includes of PETSc header files
- **Use of explicit interface blocks for type checking**
- **PETSc builds either with either real or complex types exclusively**
  - interface now implemented in *iter_cg_f90.f*
    - *fac_dir* and *solve_dir*
  - calls are within if-then(s) associated with *solver=='superlu_dist'*

Easy to test different types of solvers with PETSc

Package	Base Class	PCType	MatType	Runtime Options
DSCPACK	baij	cholesky	MATDSCPACK	-mat_type dscpack
ESSL	seqaij	lu	MATESSL	-mat_type essl
LUSOL	seqaij	lu	MATLUSOL	-mat_type lusol
MATLAB	seqaij	lu	MATMATLAB	-mat_type matlab
MUMPS	aij	lu	MATAIJMUMPS	-mat_type aijmumps
	sbaij	cholesky	MATSBAIJMUMPS	-mat_type sbaijmumps
PLAPACK	plapack	lu	MATPLAPACK	-mat_type plapack
	plapack	cholesky	MATPLAPACK	-mat_type plapack
SPOOLES	aij	lu	MATAIJSPOOLES	-mat_type aijspooles
	sbaij	cholesky	MATSBAIJSPOOLES	-mat_type sbaijspooles
SUPERLU	seqaij	lu	MATSUPERLU	-mat_type superlu
SUPERLU_DIST	aij	lu	MATSUPERLU_DIST	-mat_type superlu_dist
UMFPACK	seqaij	lu	MATUMFPACK	-mat_type umfpack

*from PETSc users manual

Really is that easy

mpirun -n 2 nimrod -mat_type superlu -pc_type lu

	CPU secs	% of total
Loop time =	9.37500E-01	6.81818E+01
Setup time =	4.37500E-01	3.18182E+01
Total time =	1.37500E+00	1.00000E+02
Iteration time =	3.12500E-01	3.33333E+01
Factoring time =	6.25000E-02	6.66667E+00

mpirun -n 2 nimrod -mat_type aijmump -pc_type lu

	CPU secs	% of total
Loop time =	1.00000E+00	6.66667E+01
Setup time =	5.00000E-01	3.33333E+01
Total time =	1.50000E+00	1.00000E+02
Iteration time =	5.00000E-01	5.00000E+01
Factoring time =	1.25000E-01	1.25000E+01

What needs to be done

- **Finish testing matrix 'triplet' interface**
  - compressed row format works for certain r-block decomposition
  - triplet interface is for distributed matrices

matrix_set_values(matrix,m,idxm,n,idxn,values,ierr)

- **Start scaling studies on larger computers**
- **A complex interface for PETSc external multigrid packages needs to be developed.**

Port to super computers is always fun

- Need to understand builds of PETSc on Bassi, Franklin and Jaguar
- Work with build system to link to appropriate libraries
- Look at using IMP and then Tau for profiling for scaling studies

Conclusion

- To ease the exploration of community provided linear solvers, we have interfaced to PETSc.
- Need to work on the complex type interfaces.
  - multigrid for peta-scaling
- Different linear solvers will be explored for large problems such as the ELM cases
- Careful study of interface latency may need to be explored.
- This can be a segue for use of PETSc's nonlinear iterative solvers.