

Lab #12: Use of PETSc

PCSE 2015

Justification

In this lab you will take a number of small PETSc examples and add functionality to them.

`init.c`

This program takes a commandline argument and prints it out.

1. Look up the routine `PetscPrintf` and use it so that the value gets printed only once.
2. Look up the routines `PetscSynchronizedPrintf` and `PetscSynchronizedFlush` and use them so that each process prints a line like 'I am process 5 out of 25', and they are printed in the proper sequence.

`mul.c`

This program constructs a vector and matrix, does the matrix-vector product, and views the result.

1. Use `VecSetValue` or `VecSetValues` to fill in the vector. Run the program. Did you get output as expected?
2. Fill in some off-diagonal elements on the matrix, for instance making a tridiagonal matrix. Use `MatSetValue` or `MatSetValues`.
3. Change one of the values in `MatMPIAIJSetPreallocation` to zero. What happens when you run the program? Do you understand the message?

sys.c

This program constructs a matrix and solves a linear system.

1. Use a call to `KSPGetConvergedReason` to print out the reason the solver terminated. Positive means success.
2. Print out how many iterations it took for the solver to converge. How does that number change if you make the linear system larger. Note: use the commandline option for changing the system size.
3. The coefficient matrix is nonsymmetric, and certain iterative methods do not work for such matrices. The function `KSPSetType` has a corresponding commandline option. Use that to specify a type of `cg`, and report on what happens.
4. Construct the residual vector and print its norm. Use `VecDuplicate`, `MatMult`, and `VecAXPY`.