CAAM 520 Randoms

Rujeko Chinomona

March 30, 2016

20 January 2016

- Go through Git Tutorial
- git checkout master
- git checkout -b knepley/feature - srmg - dm
- git : commit, push, merge, checkout next, push, branch
- *maint, master, next* : *feature branch, bugfix branch*
- **Note:** find flowchart
- Version Control for Windows and compiler or just use Ubuntu Virtual Box
- Makefile purpose :
- Make on Windows?
- Practice compiling and using make.

Test Documentation

Check CAAM 519 notes

- Read up on Latex
- Constant Coefficient Laplace - Analytic solution, Patrick Roach? Method of Manufactured Solutions
- Presentation on current progress
- Version control(group repository), Build (Makefile),
- What kind of API do we want?

22 January 2016

- git clone git@github.com:jag20/CAAM520SRM.git
- pushd
- register for git
- Automatic variables with Make
- PETSc

29 January 2016

- name mangling
- declarations
- clone git repository
- read up on required stuff
- build
- git status
- make one change

1 February 2016

- make all
-

3 February 2016

- Preconditioners
- Left and Right Preconditioners
- Jacobi
- thousand billion unknown
- multigrid - why it works
- problems with multigrid
- calculating work
- segmental refinement- memory save (run large problems on small machines)
- read up on parallelism
- potential problem in a square

8 February 2016

- Documentation
- LaTeX for paper documentation
- Function documentation
- Policies for documentation

- Kinda sorta advocating use of Sphinx
- Mathjax - A JavaScript display engine for mathematics that works in all browsers. Used on Elemental.
- Doxygen vs. Sphinx
- virtualenv...
- readthedocs documentation
- read on PCMG
- Example 5 PETSc - run with Jacobi

10 February 2016

- Using debugger
- Using nm
- did something in srmg.c
- something with type srmg worked

12 February 2016

- Method of Manufactured Solutions - test for accuracy
  - Pick an exact solution u^* and force boundary condition to agree with u^*
  - Stick in your solution $-\Delta u^* = f^*$
  - Force f to be f^*, $-\Delta \hat{u} = f^*$
  - Compare $\hat{u}$ to u^* : $\|\hat{u} - u^*\| < Ch^r$
- One problem is choosing a special solution that won't identify bugs or possible problems with code
- Plot $\log \|\hat{u} - u^*\|$ vs. h, slope r is rate of convergence
- Test Documentation - LaTeX vs. HTML
- Mathjax to put math in HTML
- Tests help build confidence with code
- Test for performance :
- Microbench-marking

22 February 2016

- Goals :
  - Build working SRMG implementation
  - Experience with team work
  - experience with structured development
- One thing to get out of failure:Declare Success
- Segmental Refinement versus Domain Decomposition
- Domain Decomposition - Solve independently on each domain and do something on boundary, eg. Schur Complement
- Segmental Refinement- Overlapping domains, Schwarz alternating procedure
- Bratu $-\Delta u + \lambda \exp(u) = 0$,
- Complicated h that breaks the algorithm : $-\nabla \cdot h(x, y)\nabla u = f$
- Drawing program - TikZ

24 February 2016

SRMG

- Bratu problem in 2D
- Example code
- Example build
- Stand in SRMG (LU, GMG,...)
- Example run (CI)
- Example verify (CI)
- Automate graphs for convergence
  - order
  - absolute disc error
  - change disc
- Travis (continuous integration program?)
- Structure code to do DFS (for binary tree - nodal dependencies)

- Charm

7 March 2016

- How to scale a code in the human dimension - Matthew J. Turk (on arxiv)
- Interact with users via mailing lists and bug trackers :
  - Humility
  - Respect
  - Trust

9 March 2016

- Coarse grid solve
- KSP Solve
- Testing
  - Bratu
  - MMS
  - Discretization error
  - Rates of convergence
- Single Patch
  - Map to patch from course grid including boundary conditions
  - $\|u - u^*\|_{patch}$
  - Single patch hierarchy
- Multiple patches (paper)
- Parallel Implementation
  - Intel parallel week April 18 - April 25

11 March 2016

- Set environment variables : PETSC_DIR, PETSC_ARCH, SRMG_DIR eg. export PETSC_DIR=/home/rue2014/petsc/
- Build SRMG library
  - git checkout next
  - git pull

  - make

- Build Example
  - cd $PETSC_DIR
  - cd src/snes/examples/tutorials
  - make ex5
  - ./ex5 -snes_view -snes_monitor -ksp_monitor -snes_view -pc_type srmg -dll_prepend $SRMG_DIR/$PETSC_ARCH/lib/libsrmg.so -options_left
  - sourcetree GUI for git for Macs and Windows
  - GUI from github for Linux

25 March 2016

- New MMS test
  a) poly MMS
- Script which finds β, c
- Higher-order interpolation item Patch convergence diagram

30 March 2016

Education Paper

Benefits of project-oriented class organized around a new research result.

Tech Paper

New experiment results for patchwise convergence of SR, evanescent memory MG

- Intro
  - State problem
  - State main result
  - Possible applications
- Background
  - Prior work
  - Open questions/problems
  - Put your work in context
- Methodology
  - Describe algorithm
  - Prook of polylog memory
  - Describe implementation
  - Patch functional description

- Results
  - Describe test cases and types of mesh
  - Plots
    * Mesh convergence MMS 1,2,3,4 (Correctness)
    * Patch convergence MMS 2,3,4 (Accuracy)
    * Memory Usage (Memory Efficient)
    * Patch convergence as a function of buffer size: constant, growing with load (Accuracy)
    * Patch convergence as a function of interpolation order (Accuracy)
    * Runtime vs. N (Efficiency)
- Conclusions
  - Restate main result
  - Open problems