Porting Machine Learning and Modeling from a Laptop to a Supercomputer: Numerical Libraries

Matthew Knepley

Portable

Extensible

Toolkit for

Scientific

computing

https://petsc.org

Earth Science with PETSc

- PyLith (CIG)
- Underworld (Monash)
- Salvus (ETHZ)
- TerraFERMA (Columbia)
- PFLOTRAN (DOE)
- STOMP (DOE)
- pTatin3d (UCSD)
- Rhea (NYU)
- Waiwera (U Auckland)

Sparse, Parallel Linear Algebra

- Vec/Mat bound to a compute resource at runtime
  "Toward Performance-Portable PETSc for GPU-based Exascale Systems"
  arXiv:2011.00715

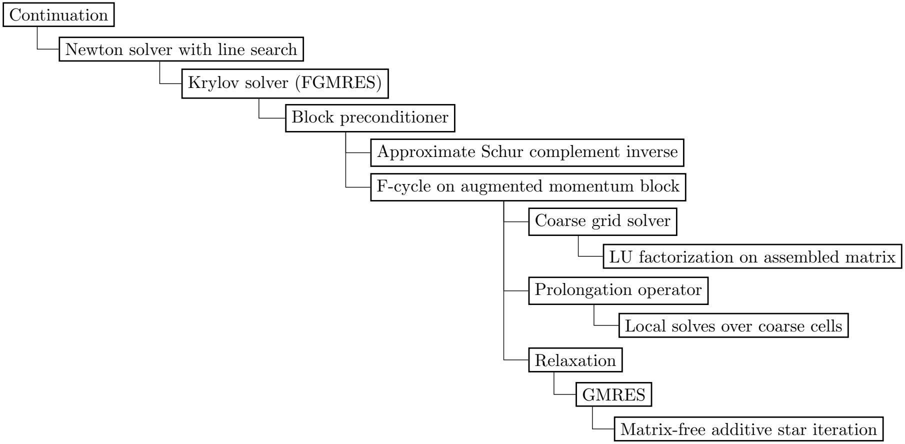
- Communication happens directly between GPUs
  "The PetscSF Scalable Communication Layer"
  arXiv:2102.13018

Implicitness

- Allow accuracy limits to supercede stability
  - CFL limits
  - Geometric limits
- Scale separation
  - Parabolization
  - Localization
- Inverse problems
  - Bayesian inference
  - Parameter estimation

Composable Solvers

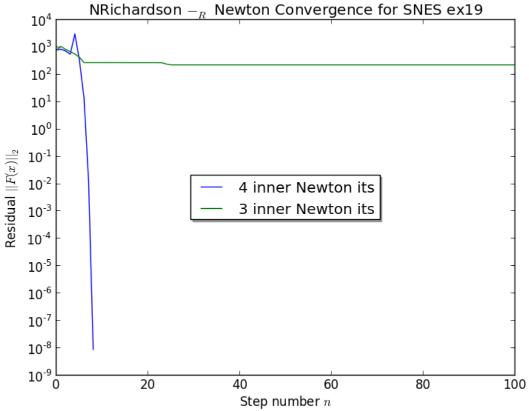
Linear


Continuation
Newton solver with line search
Krylov solver (FGMRES)
Block preconditioner
Approximate Schur complement inverse
F-cycle on augmented momentum block
Coarse grid solver
LU factorization on assembled matrix
Prolongation operator
Local solves over coarse cells
Relaxation
GMRES
Matrix-free additive star iteration

Farrell, Mitchell, and Wechsung (2018), arXiv:1810.03315

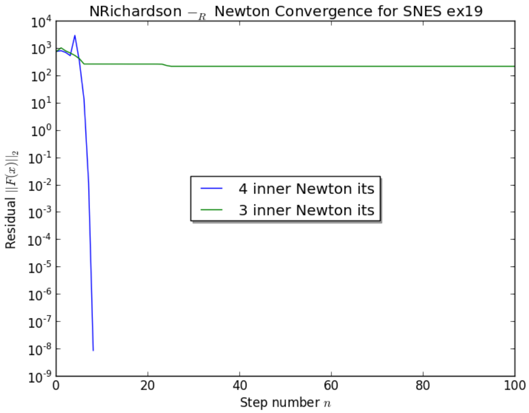
Preconditioning NRichardson with Newton

```
./ex19 -lidvelocity 100 -grashof 1.3373e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type nrichardson -snes_max_it 200
  -npc_snes_type newtonls -npc_snes_max_it 3 -npc_pc_type lu
```



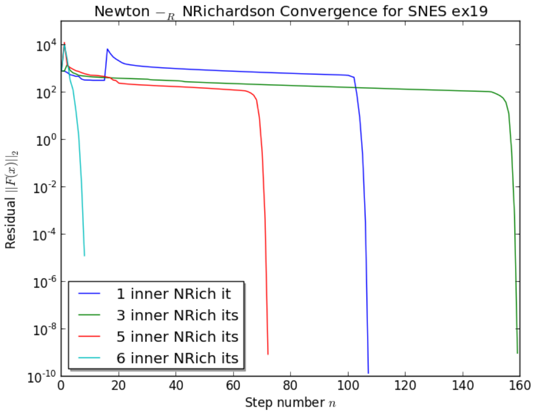
Preconditioning NRichardson with Newton

```
./ex19 -lidvelocity 100 -grashof 1.3373e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type nrichardson -snes_max_it 200
  -npc_snes_type newtonls -npc_snes_max_it 4 -npc_pc_type lu
```



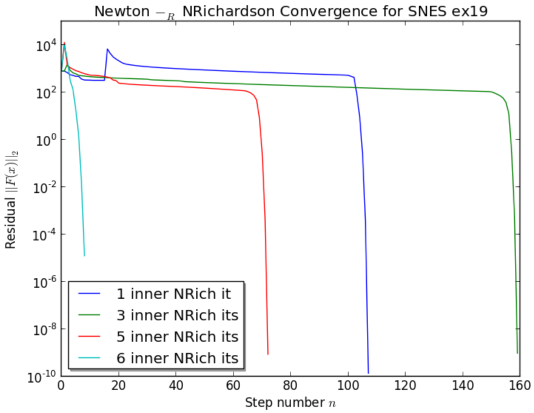
Preconditioning Newton with NRichardson

```
./ex19 -lidvelocity 100 -grashof 1.3373e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 1000 -pc_type lu
  -npc_snes_type nrichardson -npc_snes_max_it 1
```



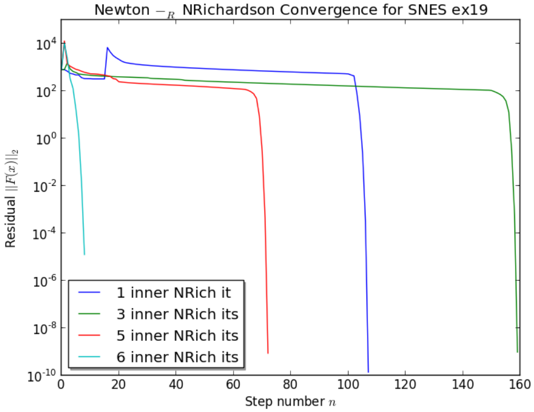
Preconditioning Newton with NRichardson

```
./ex19 -lidvelocity 100 -grashof 1.3373e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 1000 -pc_type lu
  -npc_snes_type nrichardson -npc_snes_max_it 3
```



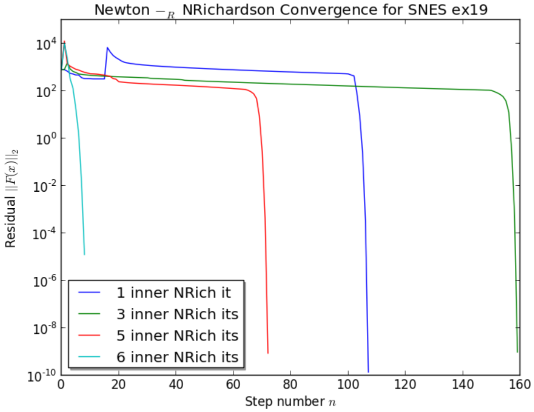
Preconditioning Newton with NRichardson

```
./ex19 -lidvelocity 100 -grashof 1.3373e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 1000 -pc_type lu
  -npc_snes_type nrichardson -npc_snes_max_it 5
```



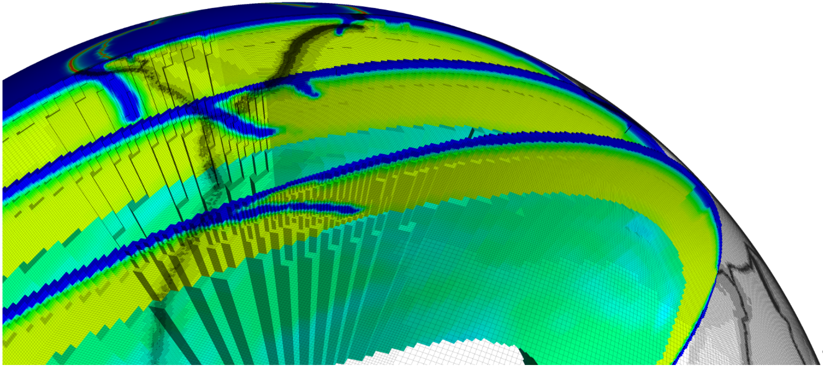
Preconditioning Newton with NRichardson

```
./ex19 -lidvelocity 100 -grashof 1.3373e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 1000 -pc_type lu
  -npc_snes_type nrichardson -npc_snes_max_it 6
```



Adaptive Mesh Refinement

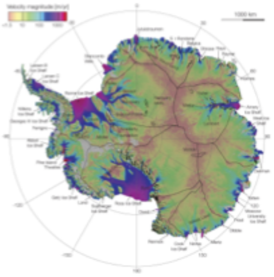
- DM interface with p4est package from Burstedde and Isaac
- PETSc solvers can be used seamlessly
- 2015 Gordon Bell Winner for Mantle Convection Simulation

[a]

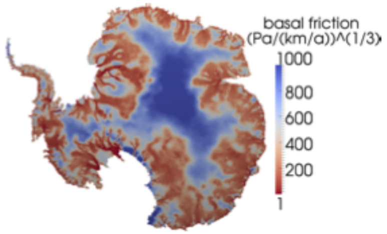
[a]Isaac

Adaptive Mesh Refinement

- DM interface with p4est package from Burstedde and Isaac
- PETSc solvers can be used seamlessly
- 2015 Gordon Bell Winner for Mantle Convection Simulation
- Inversion for basal traction on the full Antarctic ice sheet



Observed surface flow velocity (Rignot et al., 2011)

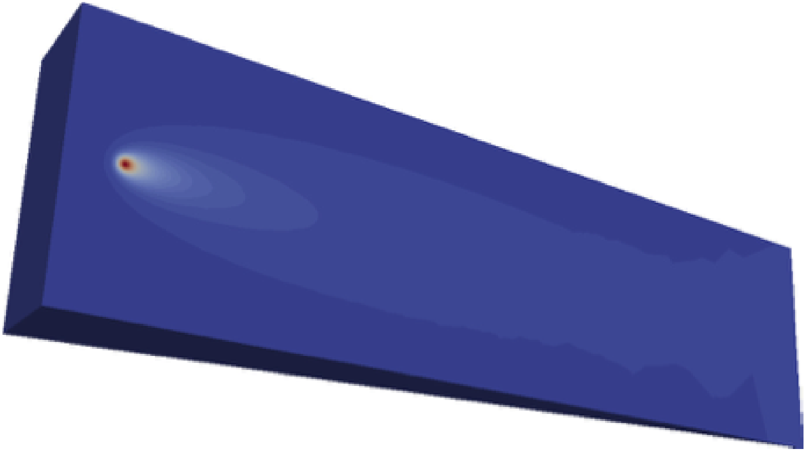


Antarctic ice sheet inversion for the basal friction parameter field using InSAR surface velocity measurements [a]

[a] Isaac

Adaptive Mesh Refinement

- DM interface with ParMmg package from Froehly and Cirrottola
- PETSc solvers can be used seamlessly
- Wallwork, Barral, Ham, Piggot (2021) eartharXiv:4205

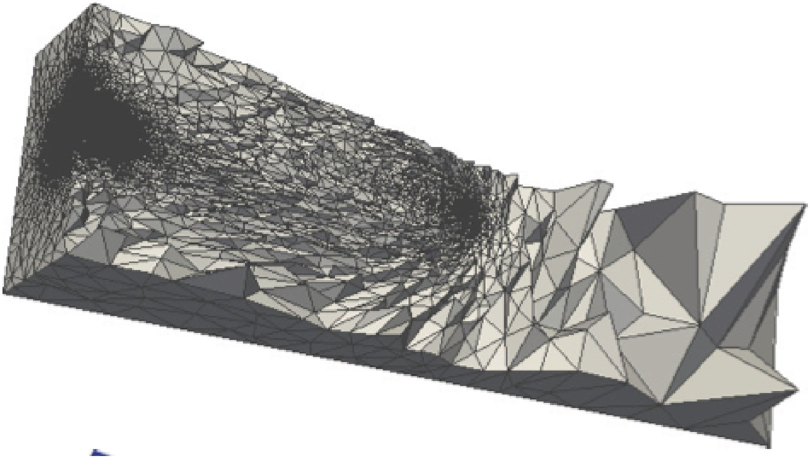
[a]

---

[a]Wallwork

Adaptive Mesh Refinement

- DM interface with ParMmg package from Froehly and Cirrottola
- PETSc solvers can be used seamlessly
- Wallwork, Barral, Ham, Piggot (2021) eartharXiv:4205


[a]

[a] Wallwork

Differentiability

- Low-level AD
  - PyTorch, Tensorflow
  - Enzyme
    “Performance-Portable Solid Mechanics via Matrix-Free *p*-Multigrid”
    arXiv:2204.01722
- High-level AD
  - pyadjoint (Firedrake, FEniCS)
    github:dolfin-adjoint/pyadjoint
  - libadjoint
    bitbucket:dolfin-adjoint/libadjoint

Why Libraries?

Libraries Hide
Hardware Details

Why Libraries?

Libraries Hide

Hardware Details

Why Libraries?

Libraries Hide
Hardware Details

Why Libraries?

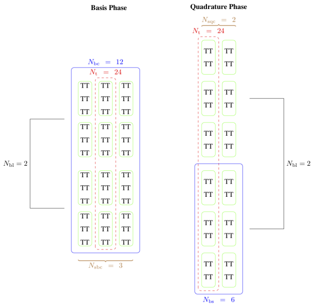
Libraries Hide Hardware Details

Why Libraries?

Libraries Hide Implementation Complexity

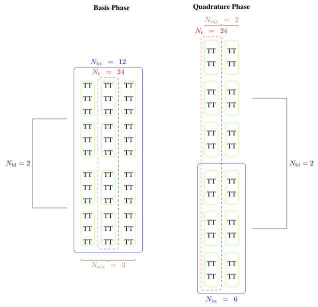
Why Libraries?

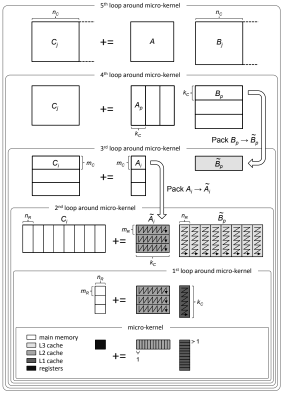
Libraries Hide Implementation Complexity



Why Libraries?

Libraries Hide Implementation Complexity

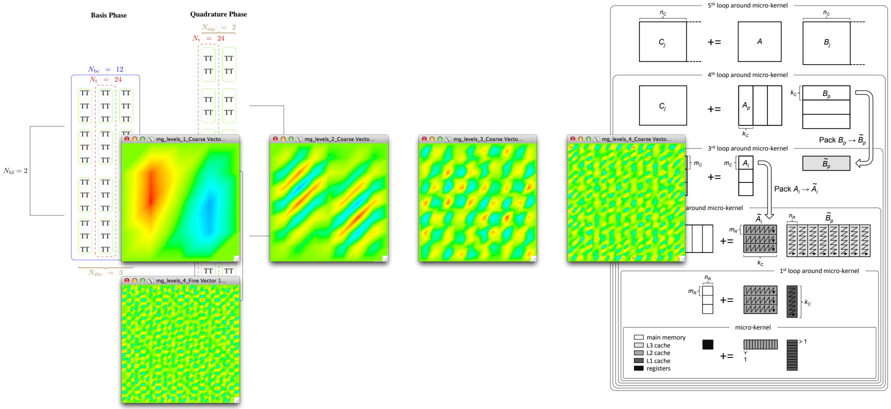




5th loop around micro-kernel
4th loop around micro-kernel
3rd loop around micro-kernel
2nd loop around micro-kernel
1st loop around micro-kernel
micro-kernel
n_C
C_j
+=
A
B_j
A_p
B_p
k_C
Pack $B_p \rightarrow \tilde{B}_p$
C_i
m_C
A_i
$\tilde{B}_p$
Pack $A_i \rightarrow \tilde{A}_i$
n_R
m_R
$\tilde{A}_i$
main memory
L3 cache
L2 cache
L1 cache
registers
1

Why Libraries?

Libraries Hide Implementation Complexity



Basis Phase
Quadrature Phase
$N_{sqc} = 2$
$N_t = 24$
$N_{bc} = 12$
$N_t = 24$
$N_{bl} = 2$
$N_{sbc} = 3$
TT
mg_levels_1_Coarse Vecto...
mg_levels_2_Coarse Vecto...
mg_levels_3_Coarse Vecto...
mg_levels_4_Coarse Vecto...
mg_levels_4_Fine Vector 1...
5th loop around micro-kernel
n_C
C_j
+=
A
B_j
4th loop around micro-kernel
A_p
B_p
k_C
Pack $B_p \rightarrow \tilde{B}_p$
3rd loop around micro-kernel
m_C
A_i
$\tilde{B}_p$
Pack $A_i \rightarrow \tilde{A}_i$
around micro-kernel
$\tilde{A}_i$
n_R
m_R
1st loop around micro-kernel
micro-kernel
main memory
L3 cache
L2 cache
L1 cache
registers
1

Why Libraries?

Libraries Accumulate
Best Practices

Libraries Accumulate Best Practices

Classical Gram-Schmidt orthogonalization with selective reorthogonalization

Libraries Accumulate Best Practices

Classical Gram-Schmidt orthogonalization with selective reorthogonalization

Eigen-estimation for AMG with Krylov bootstrap

Libraries Accumulate Best Practices

Classical Gram-Schmidt orthogonalization with selective reorthogonalization

Eigen-estimation for AMG with Krylov bootstrap

Improvement without code changes

Libraries Simplfy
Determining Provenance

Libraries Simplfy Determining Provenance

Rather than making build-time choices,

Libraries Simplfy Determining Provenance

Rather than making build-time choices, that must be plumbed through all levels

Libraries Simplfy Determining Provenance

Rather than making build-time choices,
that must be plumbed through all levels
with another layer of workflow scripts

Libraries Simplfy Determining Provenance

Rather than making build-time choices,
that must be plumbed through all levels
with another layer of workflow scripts
and brittle top-level interfaces,

Libraries Simplfy Determining Provenance

we use packages without modification,

Libraries Simplfy Determining Provenance

we use packages without modification, compiled in a standard way

Libraries Simplfy Determining Provenance

we use packages without modification,
compiled in a standard way
and controlled entirely via runtime options.

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness

- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
  - Nearly 30 years of continuous development
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness

- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
  - Tested at every supercomputer installation and 10,000+ users
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
  - Many Gordon Bell Winners
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
    - State-of-the-Art Linear and Nonlinear Solvers, Eigensolvers, Optimization Solvers, Timesteppers
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility
  - More than 8000 citations and hundreds of application packages
  - Aerodynamics, Arterial Flow, Corrosion, Combustion, Data Mining, Earthquake Mechanics, . . .