Library Development for Computational Science

Matthew Knepley $\in$ PETSc Team

Computation Institute
University of Chicago

Challenges in 21st Century Experimental Mathematical Computation
ICERM, Brown University
Providence, RI July 21–25, 2014
The PETSc Team

Bill Gropp
Barry Smith
Satish Balay

Jed Brown
Matt Knepley
Lisandro Dalcin

Hong Zhang
Mark Adams
Toby Issac
Outline

1. Computational Science
High quality numerical libraries are the best vehicle for computational communication.
The best way to create robust, efficient and scalable, maintainable scientific codes, is to use libraries.
The best way to create robust, efficient and scalable, maintainable scientific codes, is to use libraries.
The best way to create robust, efficient and scalable, maintainable scientific codes, is to use libraries.
The best way to create robust, efficient and scalable, maintainable scientific codes, is to use libraries.
Why Libraries?

- **Hides Hardware Details**
  - MPI does for this for machines and networks

- **Hide Implementation Complexity**
  - PETSc does for this Matrices and Krylov Solvers

- **Accumulates Best Practices**
  - PETSc defaults to classical Gram-Schmidt orthogonalization with selective reorthogonalization
Why Libraries?

- **Hides Hardware Details**
  - MPI does for this for machines and networks

- **Hide Implementation Complexity**
  - PETSc does for this Matrices and Krylov Solvers

- **Accumulates Best Practices**
  - PETSc defaults to classical Gram-Schmidt orthogonalization with selective reorthogonalization
Why Libraries?

- **Hides Hardware Details**
  - MPI does for this for machines and networks

- **Hide Implementation Complexity**
  - PETSc does for this Matrices and Krylov Solvers

- **Accumulates Best Practices**
  - PETSc defaults to classical Gram-Schmidt orthogonalization with selective reorthogonalization
**Why Libraries?**

- **Improvement without code changes**
  - **PETSc** time integration library has expanded rapidly, e.g. IMEX

- **Extensibility**
  - Q: Why is it not just good enough to make a fantastic working code?
  - A: Extensibility
  - Users need the ability to change your approach to fit their problem.
  - **PETSc** now does Multigrid+Block Solvers
  - **PETSc** now does Isogeometric Analysis
Why Libraries?

- **Improvement without code changes**
  - PETSc time integration library has expanded rapidly, e.g. IMEX

- **Extensibility**
  - Q: Why is it not just good enough to make a fantastic working code?
  - A: Extensibility
    - Users need the ability to change your approach to fit their problem.
  - PETSc now does Multigrid+Block Solvers
  - PETSc now does Isogeometric Analysis
Why Libraries?

- **Improvement without code changes**
  - PETSc time integration library has expanded rapidly, e.g. IMEX

- **Extensibility**
  - Q: Why is it not just good enough to make a fantastic working code?
  - A: Extensibility
    - Users need the ability to change your approach to fit their problem.
  - PETSc now does Multigrid+Block Solvers
  - PETSc now does Isogeometric Analysis
Why Libraries?

- **Improvement without code changes**
  - PETSc time integration library has expanded rapidly, e.g. IMEX

- **Extensibility**
  - **Q**: Why is it not just good enough to make a fantastic working code?
  - **A**: *Extensibility*
    - Users need the ability to change your approach to fit their problem.
  
  - PETSc now does Multigrid+Block Solvers
  - PETSc now does Isogeometric Analysis
Why Libraries?

- **Improvement without code changes**
  - PETSc time integration library has expanded rapidly, e.g. IMEX

- **Extensibility**
  - Q: Why is it not just good enough to make a fantastic working code?
  - A: Extensibility
    - Users need the ability to change your approach to fit their problem.
  - PETSc now does Multigrid+Block Solvers
  - PETSc now does Isogeometric Analysis
Why Libraries?

- Improvement without code changes
  - PETSc time integration library has expanded rapidly, e.g. IMEX

- Extensibility
  - Q: Why is it not just good enough to make a fantastic working code?
  - A: Extensibility
    Users need the ability to change your approach to fit their problem.
  - PETSc now does Multigrid+Block Solvers
  - PETSc now does Isogeometric Analysis
Early Numerical Libraries


73  EISPACK, Brian Smith et.al.

79  BLAS, Lawson, Hanson, Kincaid and Krogh

90  LAPACK, many contributors

91  PETSc, Gropp and Smith

95  MPICH, Gropp and Lusk

All of these packages had their genesis at Argonne National Laboratory/MCS