PETSc

Portable Extensible Toolkit for Scientific Computation

Alexander Wilts and Kilian Merkelbach

Content

- What is PETSc?
- What are the goals of PETSc and how are they achieved?
- What can be done with PETSc?

Portable Extensible Toolkit for Scientific Computation

- May be used on many platforms:
  - Tightly coupled systems (supercomputers)
  - Loosely coupled systems (networks)
  - GPU-Clusters
  - Individual workstations (notebook, ...)
  - Iphone
- Supports all common operating systems
  - Linux, Windows, Mac, ...

Portable **Extensible** Toolkit for Scientific Computation

- Everything is modular
  - One plugin for vectors, one for matrices, ...
  - Plugins can be exchanged or added
- Use elements from other libraries/projects
  - Possible to e.g. use different solvers than the default-PETSc-solvers
    - Create your own solvers
    - Use solvers from other libraries in your PETSc project

Portable Extensible **Toolkit** for Scientific Computation

- PETSc is a library, but not ‚just' a library
- Also integrated:
  - Debugging
  - Profiling
  - Collection of algorithms
- Enables experimentation with different algorithms/solvers/...

Portable Extensible Toolkit for **Scientific Computation**

- PETSc provides classes for:
  - Vectors and matrices
  - Distributed Arrays
  - Krylov methods
  - Nonlinear solvers
  - Preconditioners
- PETSc is used in many interesting scientific projects...
    - More on that later!

Who stands behind PETSc?

- Small team of scientists
  - developed at Argonne National Laboratory (University of Chicago)
  - 12 people, centered around Barry Smith since 1991
  - external contributors (Open Source)
- Financed by the U.S. Governemnt
  - Mainly through U.S. Deparment of Energy
- PETSc is completly free to use, even for industrial purposes!

Goals of PETSc: **Scalability**

- Scalablity:
  - The same code should run on my workstation and on a 250k core supercomputer

- PETSc is not intentended to be used for „easy“ problems!
  - Good performance on such problems is not a goal

How is **scalability** achieved?

- Shared Memory Model
  - All processors share one memory
  - Race Conditions, poor scalability
- Distributed Memory Model
  - Each processor has a private memory
  - Used by PETSc, good scalability
- Essential: Message Passing Interface(MPI)

Message Passing Interface: **MPI**

- Protocol for communication betweeen processes
  - Especially processes on different processors
- Central communicator-object manages identifier for all processes
- Language-independent
- MPI is de-facto standard for parallel programming

Goals of PETSc: **Abstraction**

- Abstraction away from the hardware
  - Allow the user to focus on mathematical problems
- Abstraction depending on the user‘s needs
  - Anywhere between high-level and low-level

Level of **abstraction** in mathematical software

- High-level mathematics interface
  - User manipulates mathematical objects
- **Algorithmic and discrete mathematics interface**
  - User manipulates mathematical objects and algorithmic objects
- Low-level computational kernels
  - User works very close to the hardware

Goals of PETSc: **Portability**

- The PETSc library is available for many languages: C, C++, FORTRAN, Python
- Code-Snippet from C:

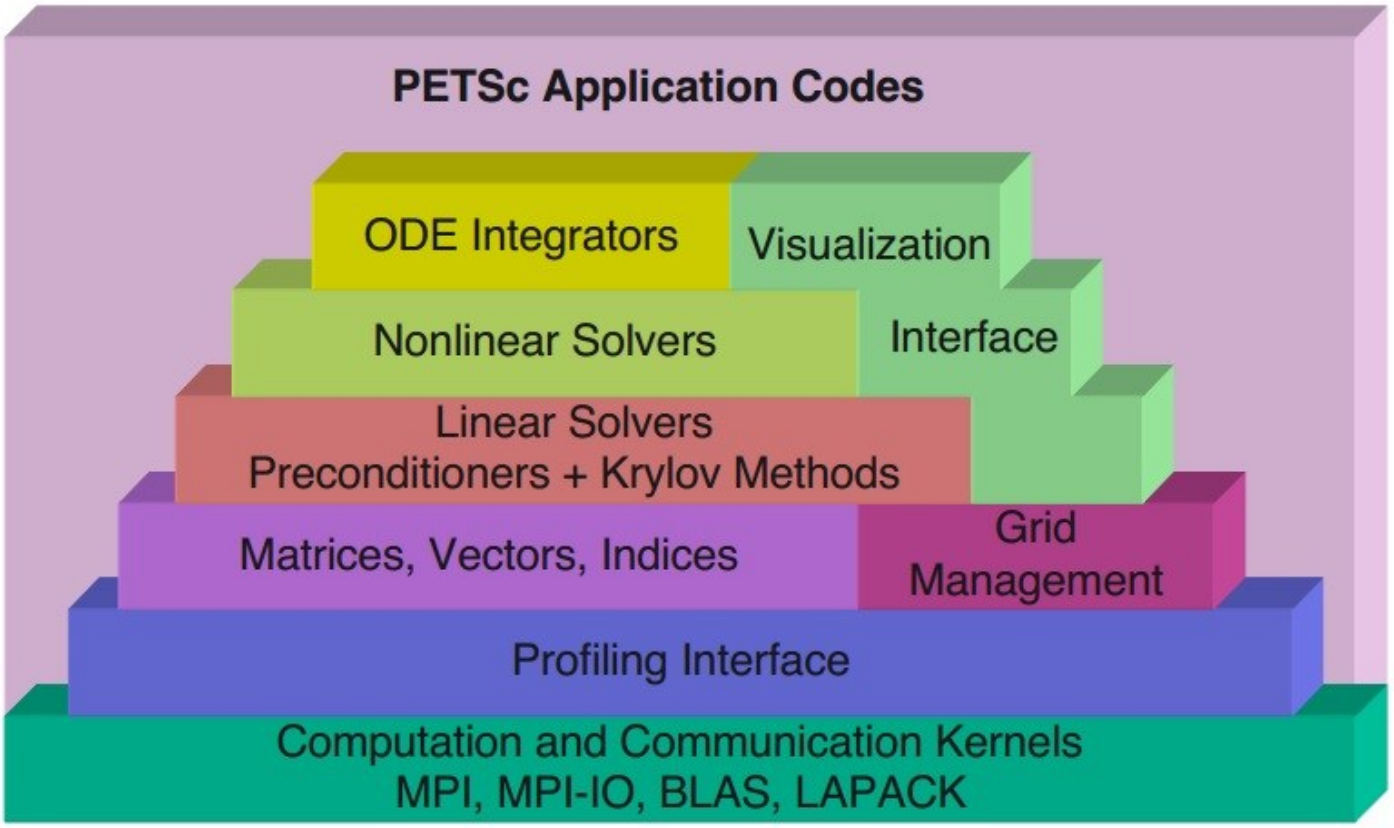
```
ierr = MatCreate(PETSC_COMM_WORLD,&A);CHKERRQ(ierr);
ierr = MatSetSizes(A,PETSC_DECIDE,PETSC_DECIDE,n,n);CHKERRQ(ierr);
ierr = MatSetFromOptions(A);CHKERRQ(ierr);
ierr = MatSetUp(A);CHKERRQ(ierr);
```

- PETSc is Object-Oriented

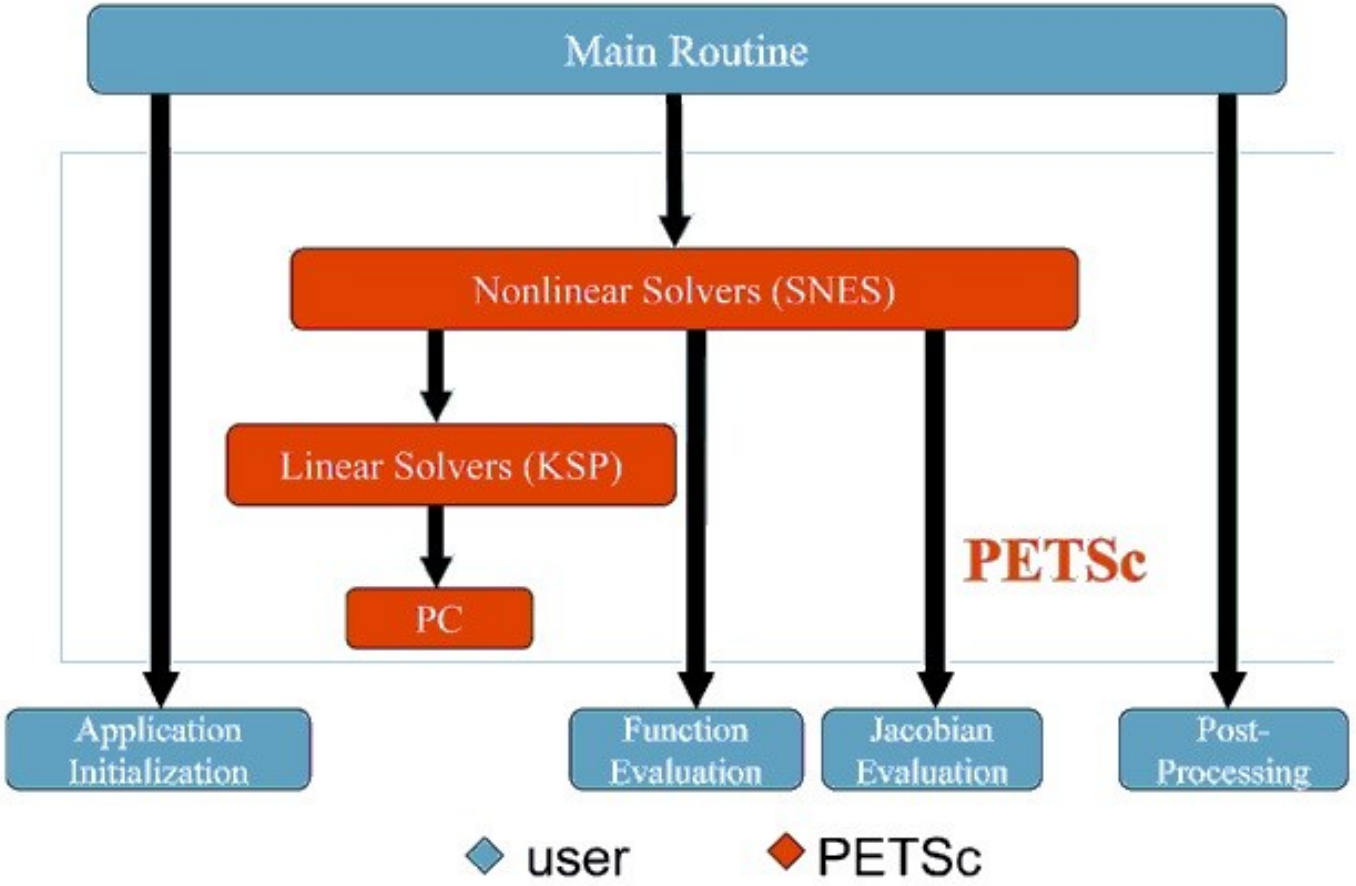
Goals of PETSc: **Experimentation**

- Main reason behind the creation of PETSc
- Rule: „It is impossible to pick the solver a priori“
- All other goals support this goal:
  - Portability: Experiment with different platforms
  - Scalability: Experiment with different hardware
  - Abstraction: Experiment with different logic

PETSc Structure Pyramid



Flow Control for a PETSc Application



Performance benchmark

- PETSc has solved problems with over 1 billion unknowns
- Runs on over 224,000 cores efficiently
- Has run at over 3 Teraflops

Source: http://www.mcs.anl.gov/petsc/documentation/tutorials/GUCASTutorial10.pdf

GPU computation

- PETSc supports computation on GPUs
  - Massive parallelism
  - Cheaper per core than CPUs
  - Up to 7x speedup compared to CPUs

Source: http://www.epcc.ed.ac.uk/sites/default/files/Dissertations/2010-2011/PramodKumbhar.pdf

PETSc and Matlab

- PETSc provides interface for Matlab
  - Interface useful for trying out different solvers and experimentation
  - But: „MATLAB is inherently not scalable“
  - Matlab more agile but PETSc more powerful

Profiling

- Built-in profiler
- Easy to activate
  - Call program with option -log_summary
- Logs FLOPS and MPI messages passed
  - Useful for experimentation

Applications

- PETSc used in a big rang of fields
  - Acoustics, Aerodynamics, Air Pollution, Arterial Flow, Bone Fractures, Brain Surgery, Cancer Surgery, Cancer Treatment, Carbon Sequestration, Cardiology, Cells, CFD, Combustion, Concrete, Corrosion, Data Mining, Dentistry, Earth Quakes, Economics, Fission, Fusion, Glaciers, Ground Water Flow, Linguistics, Mantel Convection, Magnetic Films, Material Science, Medical Imaging, Ocean Dynamics, Oil Recover, PageRank, Polymer Injection Molding, Polymeric Membrances, Quantum computing, Seismology, Semiconductors, Rockets, Relativity, Surface Water Flow

Any Questions?

...actually please don‘t ask any questions.