Metrics of success for LAPACK and <T>LAPACK

Weslley da Silva Pereira Julien Langou

University of Colorado Denver

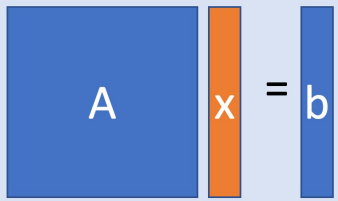
Metrics2023 – Measuring Success of Cyber Infrastructure Projects – November 12, 2023

Numerical Linear Algebra software

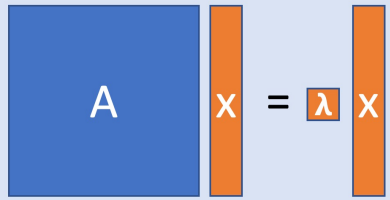
(fast + stable) + (interoperable) + (portable)

Problems to be solved

Linear system of equations



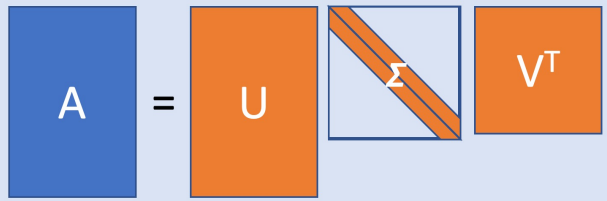
Eigenvalues



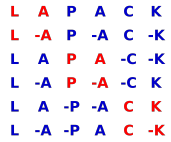
Linear Least Squares

$$\min_x \| b - A x \|_2$$

Singular Value Decomposition



LAPACK and <T>LAPACK

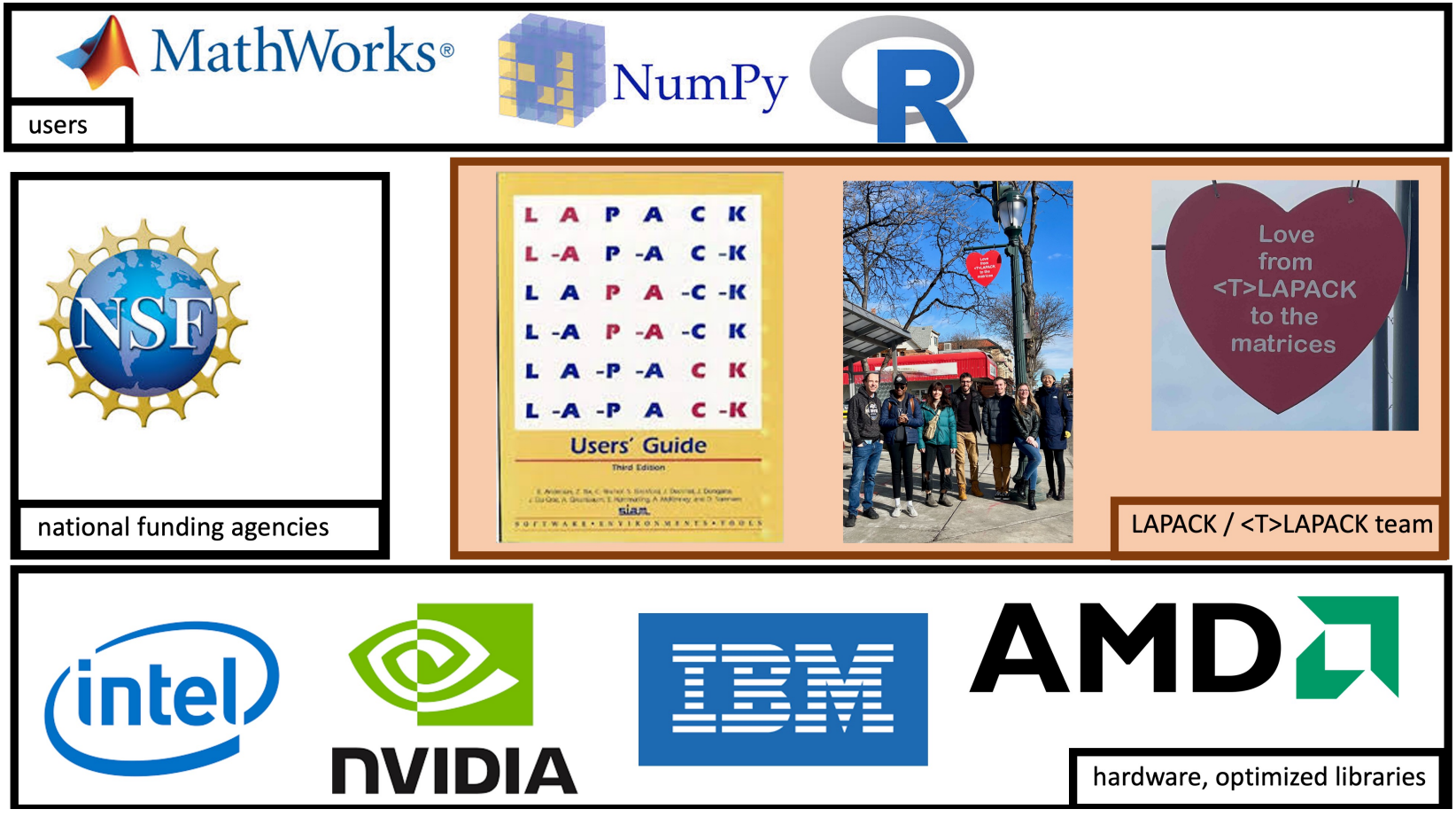


1. LAPACK is a reference legacy software written primarily in Fortran, first release in 1992, initially the main PIs were Jack Dongarra and Jim Demmel, takes a lots of inspiration from EISPACK (1976) and LINPACK (1979), based on BLAS (1979)
2. <T>LAPACK is a new software written in C++
   - Routines use high-level abstraction for matrices and types.
   - Interoperability with: SLATE, mdspan, Eigen3, StarPU.

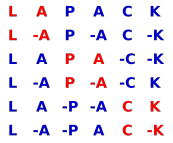
```
1 // <T>LAPACK User's code looks like:
2
3 kokkos::mdspan<double,dextents<int,2>> A( ptr, 100, 100 );
4 Eigen::MatrixXd B = Eigen::MatrixXd::Random(100,100);
5
6 int infoA = potrf( upper_triangle, A );
7 int infoB = potrf( upper_triangle, B );
```

```
1 // Inside <T>LAPACK's potrf we find commands like:
2
3 auto AJJ = slice( A, range{j,j+nb}, range{j,j+nb} );
4 AJJ(0,0) = sqrt( real( AJJ(0,0) ) );
```

Ecosystem for the LAPACK and <T>LAPACK libraries



What is there to do in a 30-year-old software?



- Stay in close contact within our ecosystem partners, maintain a community of users.
- Still very much used
- Numerical Linear Algebra is still an active field of research and new algorithms are being designed every so often. We strive to have a quick turn around between the publication of a new algorithm and its introduction in LAPACK. Examples: Early Aggressive Deflation (QR and QZ), Jacobi SVD algorithm, Communication Avoiding Algorithms, MRRR, CS Decomposition, etc.
- new directions such as: new precisions, batched BLAS, exception handling, etc.
- keep up with porting software to new machines, new installation systems

Metrics for success of LAPACK

- **# of citations.** LAPACK Users' Guide 3rd Edition (1999) receives between 300 and 500 citations every year.
- **# first-time contributors.**
  7 out of 15 contributors in LAPACK 3.10.1 (2022).
  10 out of 19 contributors in LAPACK 3.11.0 (2022).
  23 out of 42 contributors in LAPACK 3.12.0 (coming soon).
- **GitHub metrics:**
  55 PRs accepted 25 issues closed for LAPACK 3.10.
  70 PRs accepted 61 issues closed for LAPACK 3.11.
  There is still work to do: 84 issues open, 11 PRs ready.
  1.3k stars and 395 forks on Github.
- **training of students and training of postdocs:**
  about 5 undergrads, 2 grads, 1 postdoc per year.

More metrics... still to be quantified

- **Usability / maintainability:**
  Windows x Linux x MacOS.
  Feedback from collaborators and students.
- **Integration with third-party software:**
  How is the interface designed to accommodate upgrades?
- **Documentation:**
  # of documented code entities / # total number of entities.
  How accurate the documentation is?
- **Quality/security of the software:**
  OpenSSF scorecard. 6.7 for LAPACK, 8.3 for <T>LAPACK.
  Conform with security guidelines from national institutions.
  # of issues the CI can catch / # total issues found.
  Coverage of the tests. (complicated for C++ templates).
- **Impact on other software:**
  Eigen, GCC, StarPU, SLATE project, BLAS and LAPACK.


Love
from
<T>LAPACK
to the
matrices