

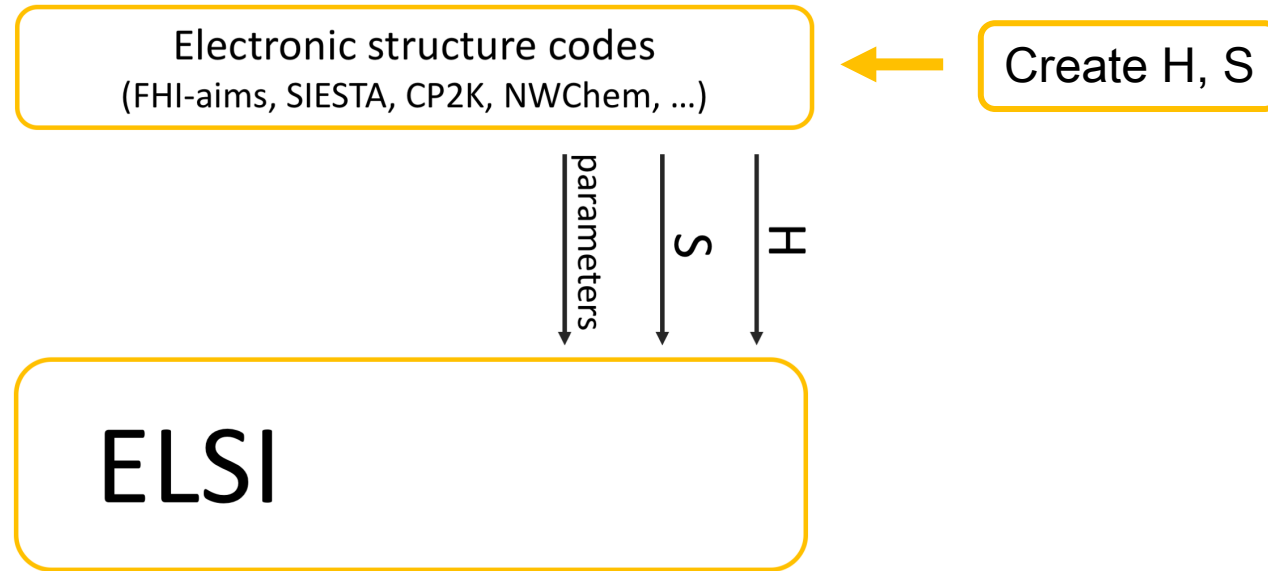
ELSI Interface Status Update

Victor Yu

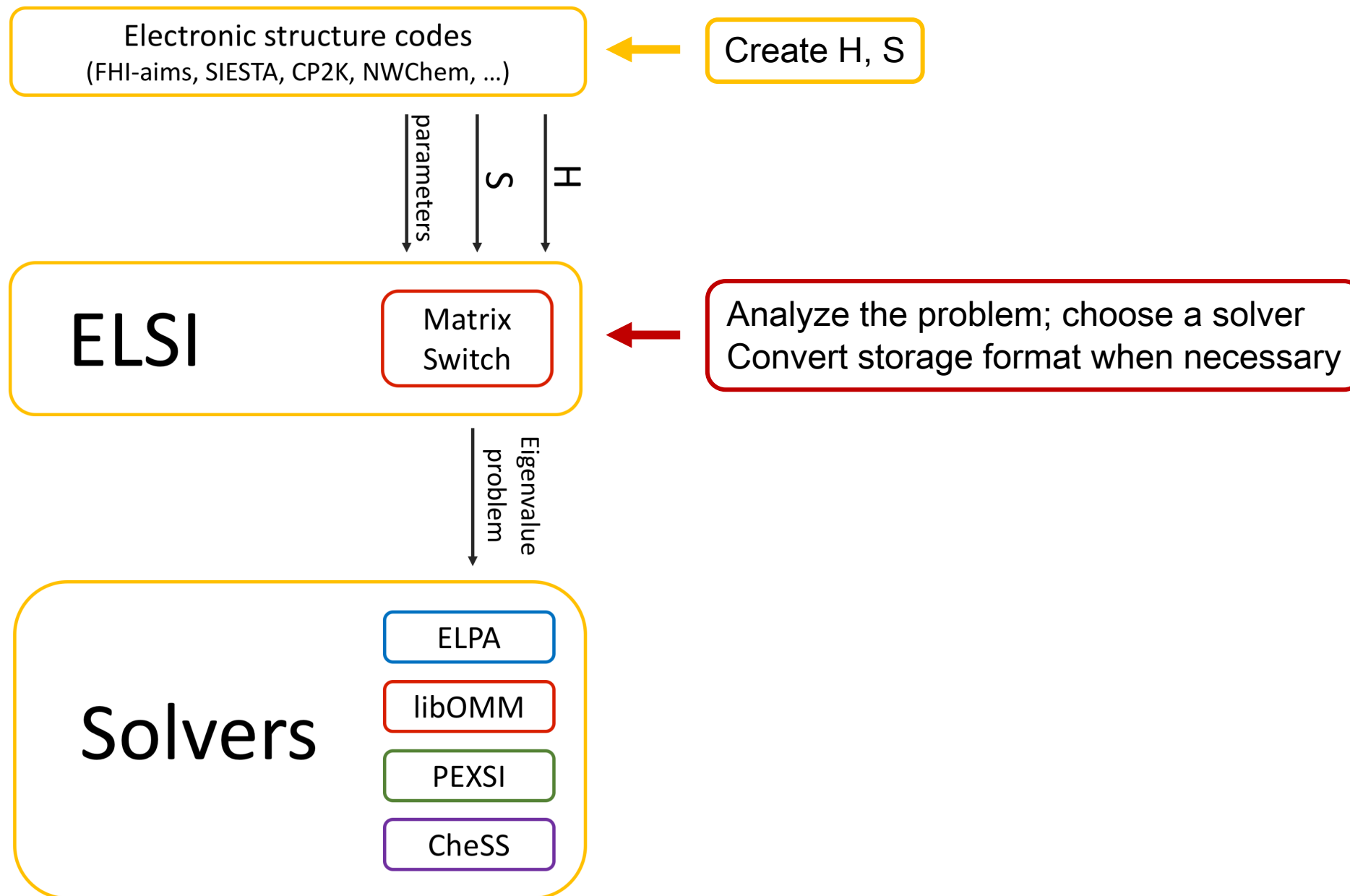
Electronic Structure Infrastructure (ELSI): Basic Ideas

Electronic structure codes
(FHI-aims, SIESTA, CP2K, NWChem, ...)

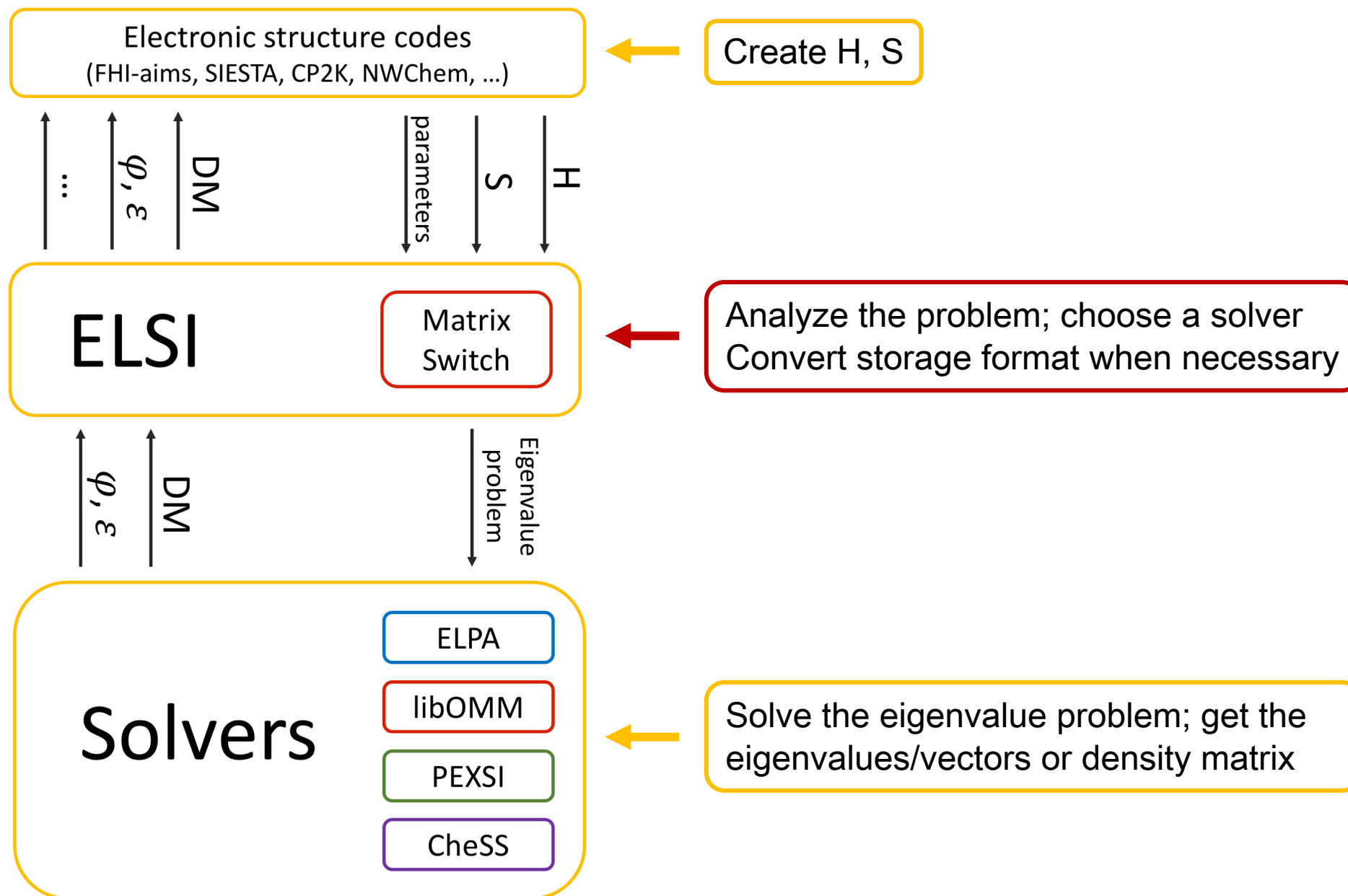
Electronic Structure Infrastructure (ELSI): Basic Ideas



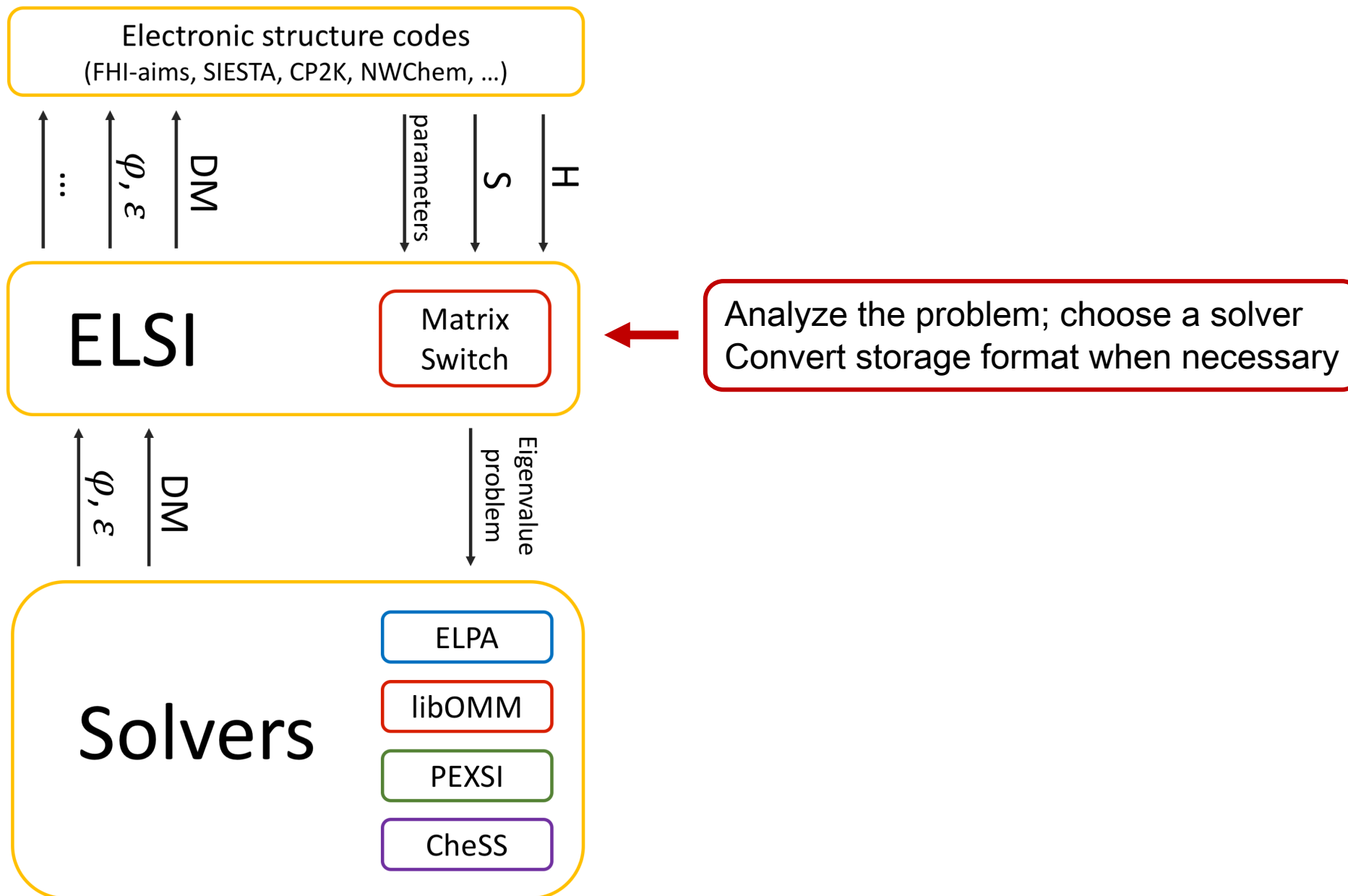
Electronic Structure Infrastructure (ELSI): Basic Ideas



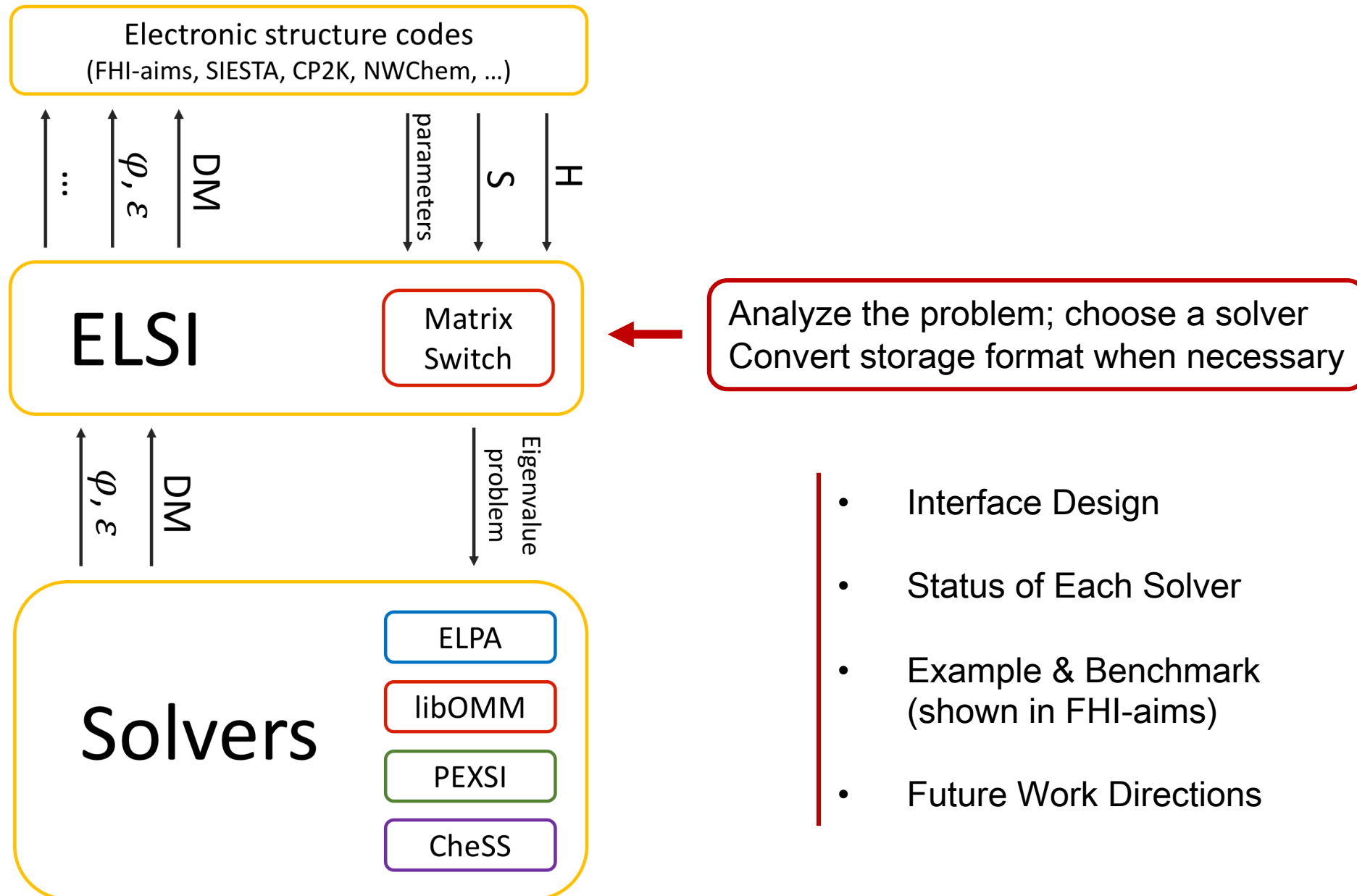
Electronic Structure Infrastructure (ELSI): Basic Ideas



Outline



Outline



ELSI Interface Design: Goals

- **Easy to implement**

No must-know for solver libraries

No more information needed by ELSI compared to ScaLAPACK

- **Versatile**

Compute the energy, density matrix, and eigensystem

- **Flexible**

All technical settings in solver libraries are adjustable for advanced users

ELSI Interface Design: Difficulties

- **Easy to implement**

No must-know for solver libraries

No more information needed by ELSI compared to ScaLAPACK

- **Versatile**

Compute the energy, density matrix, and eigensystem

- **Flexible**

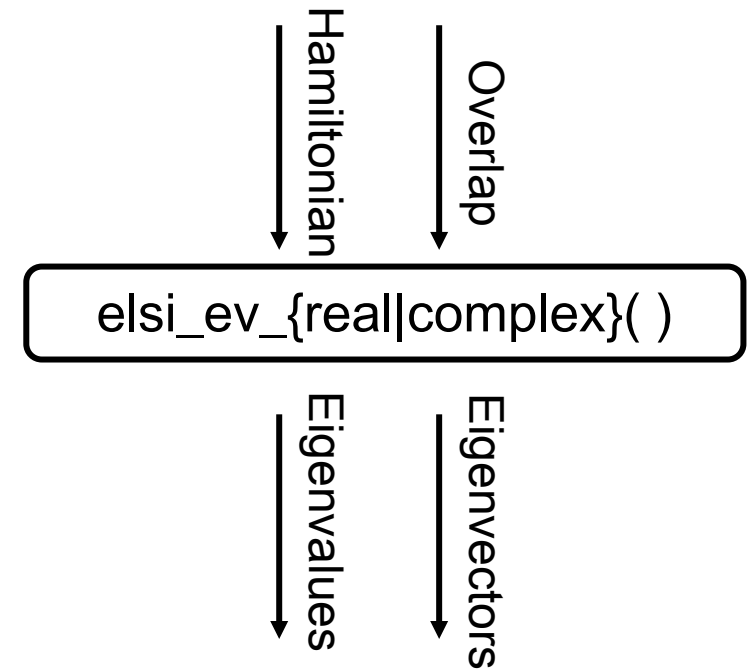
All technical settings in solver libraries are adjustable for advanced users

- How to set reasonable default settings for users who know nothing about the external solvers?
- How to keep the interface clean without losing any flexibility?

ELSI Interface Design

elsi_ev_{real|complex}: eigenvalue/eigenvector solver

- Supported method: ELPA (1-stage, 2-stage)
- Input: real or complex Hamiltonian and overlap matrices + parameters
- Output: eigenvalues and eigenvectors



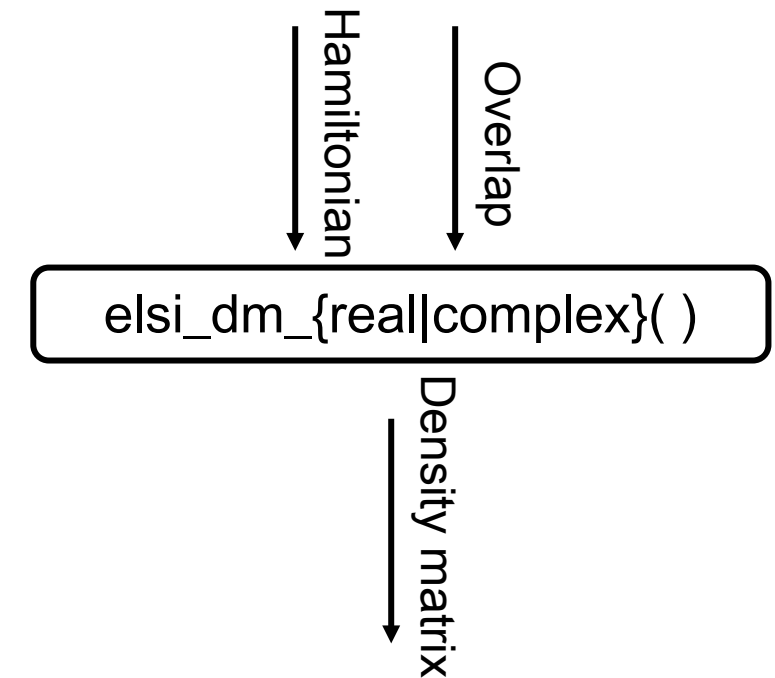
ELSI Interface Design

elsi_ev_{real|complex}: eigenvalue/eigenvector solver

- Supported method: ELPA (1-stage, 2-stage)
- Input: real or complex Hamiltonian and overlap matrices + parameters
- Output: eigenvalues and eigenvectors

elsi_dm_{real|complex}: density matrix solver

- Supported methods: ELPA, libOMM, PEXSI, CheSS
- Input: real or complex Hamiltonian and overlap matrices + parameters (+ occupation numbers)
- Output: density matrix



- Set method
(automatic decision)
- Set matrix format
- Set matrix size



ELSI used in a normal self-consistent field (SCF) solver routine

```
elsi_init ( method, format, dimension )
```

```
elsi_set_mpi ( communicators, ... )
```

```
elsi_set_blacs ( parameters )
```

```
! SCF loop
```

```
    elsi_customize_{elpa|omm|pexsi} ( parameters=choice )
```

```
    elsi_{dm|ev}_{real|complex} ( H, S, D or ev )
```

```
! End SCF loop
```

```
elsi_finalize ( )
```

Pass MPI and BLACS
information into ELSI



ELSI used in a normal self-consistent field (SCF) solver routine

```
elsi_init ( method, format, dimension )
```

```
elsi_set_mpi ( communicators, ... )
```

```
elsi_set_blacs ( parameters )
```

```
! SCF loop
```

```
    elsi_customize_{elpa|omm|pexsi} ( parameters=choice )
```

```
    elsi_{dm|ev}_{real|complex} ( H, S, D or ev )
```

```
! End SCF loop
```

```
elsi_finalize ( )
```

Override default settings of
the chosen solver (optional)



ELSI used in a normal self-consistent field (SCF) solver routine

```
elsi_init ( method, format, dimension )
```

```
elsi_set_mpi ( communicators, ... )
```

```
elsi_set_blacs ( parameters )
```

```
! SCF loop
```


```
    elsi_customize_{elpa|omm|pexsi} ( parameters=choice )
```

```
    elsi_{dm|ev}_{real|complex} ( H, S, D or ev )
```

```
! End SCF loop
```

```
elsi_finalize ( )
```

Call ELSI density matrix
solver or eigensystem solver



ELSI used in a normal self-consistent field (SCF) solver routine

```
elsi_init ( method, format, dimension )
```

```
elsi_set_mpi ( communicators, ... )
```

```
elsi_set_blacs ( parameters )
```

```
! SCF loop
```

```
    elsi_customize_{elpa|omm|pexsi} ( parameters=choice )
```

```
    elsi_{dm|ev}_{real|complex} ( H, S, D or ev )
```

```
! End SCF loop
```

```
elsi_finalize ( )
```

- Free memory
- Print timings & summary



ELSI used in a normal self-consistent field (SCF) solver routine

```
elsi_init ( method, format, dimension )
```

```
elsi_set_mpi ( communicators, ... )
```

```
elsi_set_blacs ( parameters )
```

```
! SCF loop
```

```
    elsi_customize_{elpa|omm|pexsi} ( parameters=choice )
```

```
    elsi_{dm|ev}_{real|complex} ( H, S, D or ev )
```

```
! End SCF loop
```

```
elsi_finalize ( )
```


ELPA Status

DONE

- ELPA 2015.11.001 in ELSI package can be installed with Makefile.elsi
- Solve a series of eigenvalue problems within the SCF cycle using ELPA 1-stage or 2-stage solver, real (tested) or complex (not carefully tested yet) case
- Construct density matrix from eigenvectors and occupation numbers (tested; results identical to OMM)
- Customize subroutine to choose 1-stage or 2-stage solver
- Benchmark

ELPA Status

DONE

- ELPA 2015.11.001 in ELSI package can be installed with Makefile.elsi
- Solve a series of eigenvalue problems within the SCF cycle using ELPA 1-stage or 2-stage solver, real (tested) or complex (not carefully tested yet) case
- Construct density matrix from eigenvectors and occupation numbers (tested; results identical to OMM)
- Customize subroutine to choose 1-stage or 2-stage solver
- Benchmark

TODO

- Support 3 versions of ELPA: stable release (2016.05.003), GPU, and development
- Finish the complex case

OMM Status

DONE

- OMM in ELSI package can be installed with `Makefile.elsi`
- Use Cholesky factorization in ELPA and OMM flavour 2 to solve one eigenvalue problem (density matrix identical to ELPA)
- Customize subroutine to set OMM parameters
- Proof-of-concept benchmark

OMM Status

DONE

- OMM in ELSI package can be installed with Makefile.elsi
- Use Cholesky factorization in ELPA and OMM flavour 2 to solve one eigenvalue problem (density matrix identical to ELPA)
- Customize subroutine to set OMM parameters
- Proof-of-concept benchmark

TODO

- Full test within an actual SCF cycle (FHI-aims, SIESTA)
- Sparse version of OMM

PEXSI Status

DONE

- PEXSI v0.9.2 in ELSI package can be installed with Makefile.elsi
- Format transformation: block-cyclic to distributed CCS used in PEXSI
- Solve one eigenvalue problem using PEXSI, with most parameters set by hand

PEXSI Status

DONE

- PEXSI v0.9.2 in ELSI package can be installed with Makefile.elsi
- Format transformation: block-cyclic to distributed CCS used in PEXSI
- Solve one eigenvalue problem using PEXSI, with most parameters set by hand

TODO

- Format transformation: distributed CCS to block-cyclic
- Reasonable default settings
- Benchmark
- Towards auto-decision

CHESS Status

CheSS: CHEbyshev Sparse Solvers (launchpad.net/chess)

Author: Stephan Mohr (BigDFT team)

Features:

- High level operations based on **Chebyshev expansions** for **sparse** matrices
- An implementation of the **Fermi Operator Expansion** method to calculate the **density matrix** in electronic structure calculations
- An efficient way of calculating the power of a sparse matrix

CHESS Status

CheSS: CHEbyshev Sparse Solvers (launchpad.net/chess)

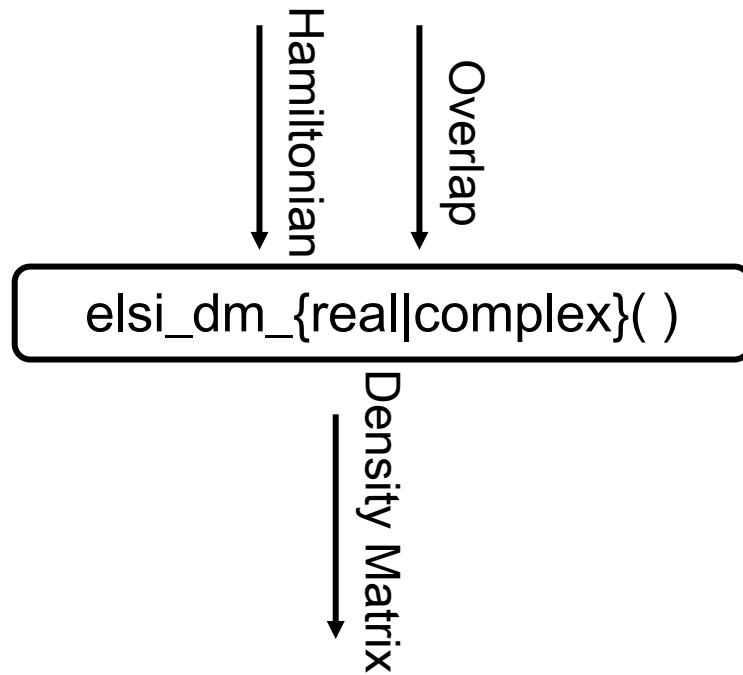
Author: Stephan Mohr (BigDFT team)

Features:

- High level operations based on **Chebyshev expansions** for **sparse** matrices
- An implementation of the **Fermi Operator Expansion** method to calculate the **density matrix** in electronic structure calculations
- An efficient way of calculating the power of a sparse matrix

TODO

- CheSS sparse data format is being implemented into MatrixSwitch
- Put CheSS source code in ELSI



FHI-aims + ELSI (solve_KS_eigen_elsi.f90)

```
call elsi_init ( elsi_method, elsi_storage, n_basis, n_states )
call elsi_set_mpi ( ... )
call elsi_set_blacs ( ... )

...

call elsi_dm_real ( ham(:, :, i_spin), ovlp, dm_test, energy, &
                  factor_overlap, occ )

...

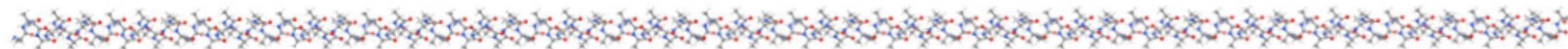
call elsi_finalize ( )
```

Benchmark: ELSI in FHI-aims

- **Example system**

a) Helical poly-alanine molecule “a200” (2003 atoms);

b) Melon (polymerized heptazine) 4-layer model “m4” (288 atoms).

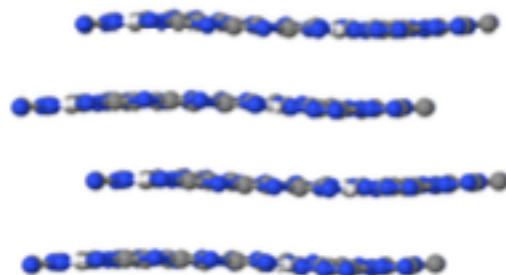


“a200”

- “tier 1” (19,024 basis functions)
- “tier 2” (54,069 basis functions)
- “tier 3” (86,117 basis functions)

“m4”

- “tier 1” (3,600 basis functions)
- “tier 2” (10,080 basis functions)
- “tier 3” (14,688 basis functions)
- “tier 4” (20,688 basis functions)
- “further” (24,144 basis functions)



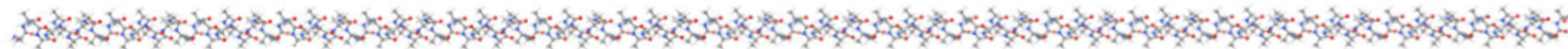
Benchmark: ELSI in FHI-aims

- **Example system**

- a) Helical poly-alanine molecule “a200” (2003 atoms);
- b) Melon (polymerized heptazine) 4-layer model “m4” (288 atoms).

- **Electronic structure method**

DFT-PBE, FHI-aims all-electron code, numeric atom-centered orbitals with variable basis set size.

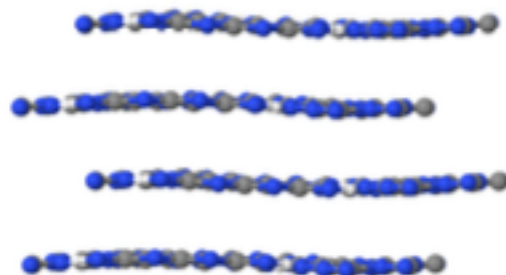


“a200”

- “tier 1” (19,024 basis functions)
- “tier 2” (54,069 basis functions)
- “tier 3” (86,117 basis functions)

“m4”

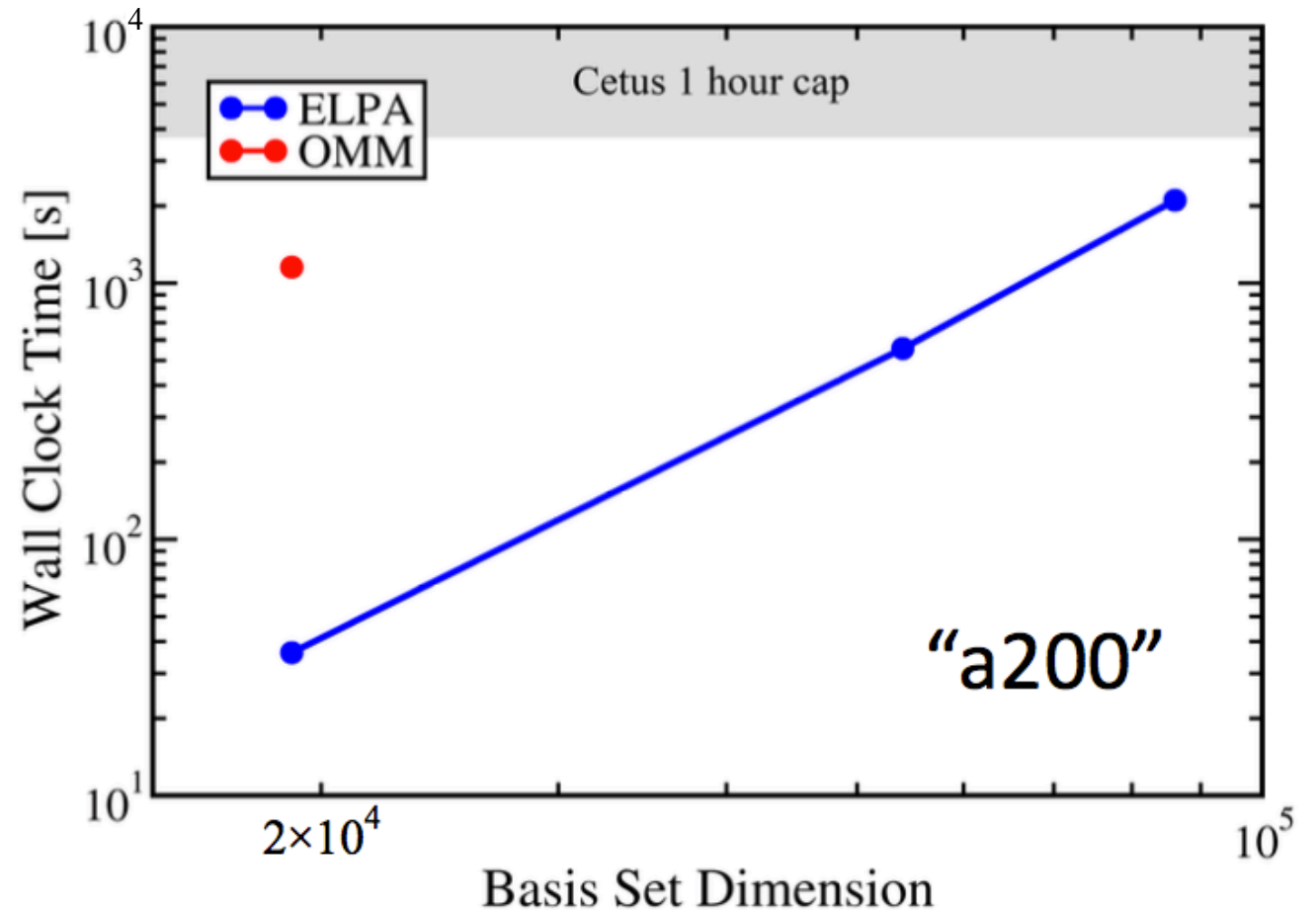
- “tier 1” (3,600 basis functions)
- “tier 2” (10,080 basis functions)
- “tier 3” (14,688 basis functions)
- “tier 4” (20,688 basis functions)
- “further” (24,144 basis functions)



1024 nodes
1 MPI task/node
Cetus @ANL
IBM Blue Gene/Q

Benchmark: ELSI in FHI-aims

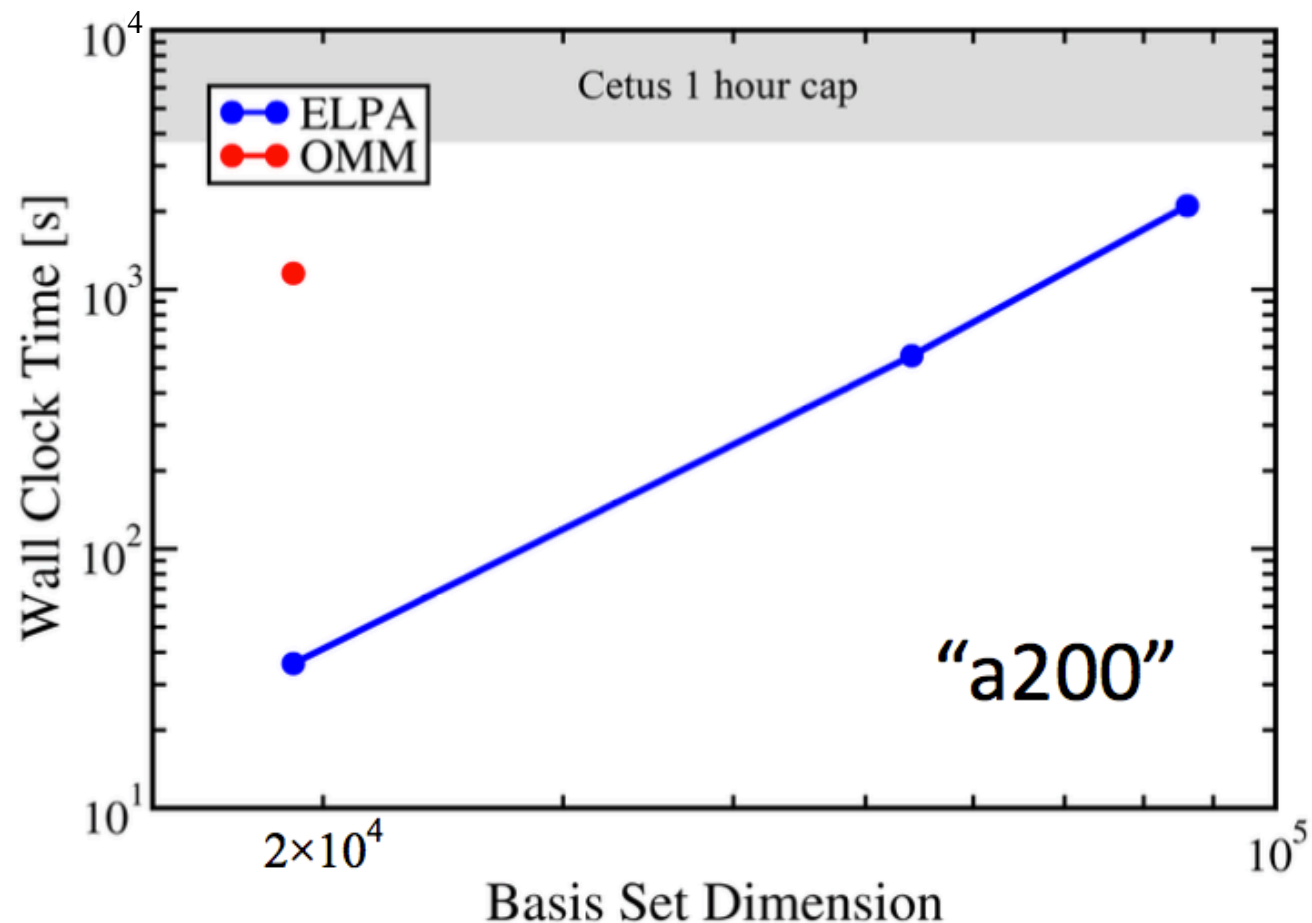
Timing of ELPA and OMM solvers in ELSI used to compute the **density matrix** from H and S constructed by FHI-aims in the **first** self-consistent field (SCF) cycle.



Benchmark: ELSI in FHI-aims

- Successful demonstration of the connection of existing electronic structure code FHI-aims ↔ ELSI interface ↔ ELPA and OMM.
- ELPA first computes the eigenvalues and eigenvectors, then constructs the density matrix.
- Flavour 2 of LibOMM is used with Cholesky factorization by ELPA.
- LibOMM is designed to benefit from efficient reuse of information within an SCF loop and/or among multiple spins/k points. Shown here is proof of concept only.

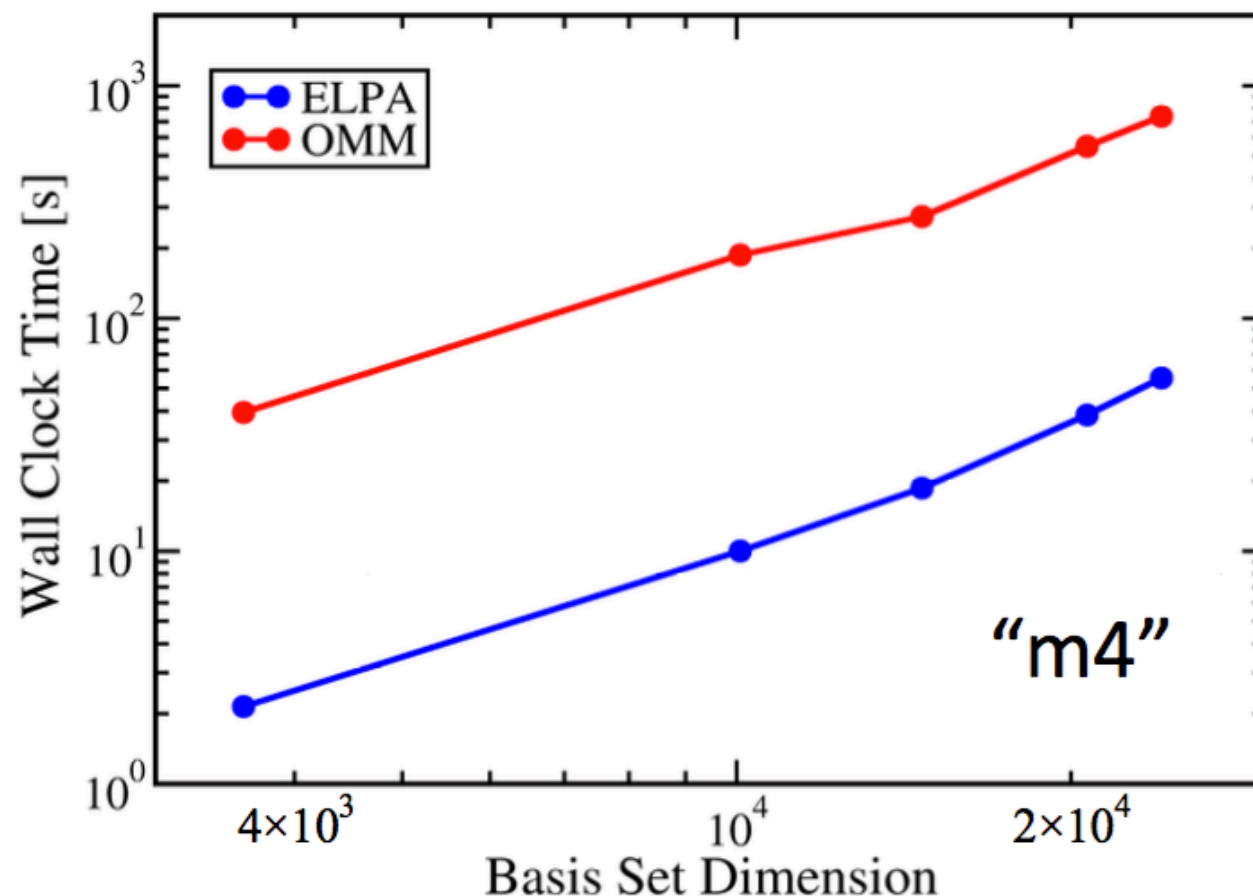
Timing of ELPA and OMM solvers in ELSI used to compute the **density matrix** from H and S constructed by FHI-aims in the **first** self-consistent field (SCF) cycle.



Benchmark: ELSI in FHI-aims

- Successful demonstration of the connection of existing electronic structure code FHI-aims ↔ ELSI interface ↔ ELPA and OMM.
- ELPA first computes the eigenvalues and eigenvectors, then constructs the density matrix.
- Flavour 2 of LibOMM is used with Cholesky factorization by ELPA.
- LibOMM is designed to benefit from efficient reuse of information within an SCF loop and/or among multiple spins/k points. Shown here is proof of concept only.

Timing of ELPA and OMM solvers in ELSI used to compute the **density matrix** from H and S constructed by FHI-aims in the **first** self-consistent field (SCF) cycle.



Future Work

- **ELPA**
 - 1) Complete the support to 3 versions of ELPA: stable release, GPU, and development
 - 2) Finish the tests of ELPA complex solver

Future Work

- **ELPA**

- 1) Complete the support to 3 versions of ELPA: stable release, GPU, and development
- 2) Finish the tests of ELPA complex solver

- **OMM**

- 1) Modify FHI-aims and test OMM self-consistently
- 2) Start implement the sparse version of OMM in ELSI (Fabiano, Haizhao)

Future Work

- **ELPA**

- 1) Complete the support to 3 versions of ELPA: stable release, GPU, and development
- 2) Finish the tests of ELPA complex solver

- **OMM**

- 1) Modify FHI-aims and test OMM self-consistently
- 2) Start implement the sparse version of OMM in ELSI (Fabiano, Haizhao)

- **PEXSI (@ Berkeley)**

- 1) Optimize the format transformation in both directions
- 2) Find best default settings for the parameters

Future Work

- **ELPA**

- 1) Complete the support to 3 versions of ELPA: stable release, GPU, and development
- 2) Finish the tests of ELPA complex solver

- **OMM**

- 1) Modify FHI-aims and test OMM self-consistently
- 2) Start implement the sparse version of OMM in ELSI (Fabiano, Haizhao)

- **PEXSI (@ Berkeley)**

- 1) Optimize the format transformation in both directions
- 2) Find best default settings for the parameters

- **ELSI**

- 1) Substitute the HDF5 regression test with the new ELSI test suite (Fabiano)
- 2) New webpage design
- 3) Installer based on GNU autotools

THANKS!

wy29@duke.edu

Git access: elsi-team@duke.edu