Recent Advancements in ELPA: Best Practices in Real Applications

Danilo Simoes Brambila and Christian Carbogno

Fritz-Haber-Institut der Max-Planck-Gesellschaft, Berlin - Germany
ELPA-AEO is an *Application-Driven Team-Effort*

See [http://elpa-aeo.mpcdf.mpg.de](http://elpa-aeo.mpcdf.mpg.de) for a full list of contributors

---

**Code Testing, Benchmarking, and Validation in *Real Applications***

*Fritz Haber Institute*
ELPA-AEO is an Application-Driven Team-Effort

See [http://elpa-aeo.mpcdf.mpg.de](http://elpa-aeo.mpcdf.mpg.de) for a full list of contributors

### Electronic Structure Theory
Density Functional Theory

### Solid State Physics & Material Science

**Thermoelectric Clathrates**

**Thermoelectric Boron Carbides**

Code Testing, Benchmarking, and Validation in Real Applications

Fritz Haber Institute

Other applications include...

- Interstitials
- Regular sites
- Thermoelectric Boron Carbides
ELPA-AEO is an Application-Driven Team-Effort

See [http://elpa-aeo.mpcdf.mpg.de](http://elpa-aeo.mpcdf.mpg.de) for a full list of contributors

---

**Electronic Structure Theory**
- Density Functional Theory

---

**Solid State Physics & Material Science**

**Thermoelectric Clathrates**

**Free Energy Barriers for H\(_2\)O on TiO\(_2\)**

---

**Quantum Chemistry**

**Occupations Disorder in Li\(_4\)Ti\(_5\)O\(_{12}\) Batteries**

---

**Code Testing, Benchmarking, and Validation in Real Applications**

---

**Functional Boron Carbides**

---

**Fritz Haber Institute**

Christoph Scheurer
ELPA-AEO is an Application-Driven Team-Effort

See [http://elpa-aeo.mpcdf.mpg.de](http://elpa-aeo.mpcdf.mpg.de) for a full list of contributors
ELPA-AEO is an Application-Driven Team-Effort

See [http://elpa-aeo.mpcdf.mpg.de](http://elpa-aeo.mpcdf.mpg.de) for a full list of contributors.
ELPA-AEO is an *Application-Driven Team-Effort*

See [http://elpa-aeo.mpcdf.mpg.de](http://elpa-aeo.mpcdf.mpg.de) for a full list of contributors
ELPA-AEO is an Application-Driven Team-Effort

See [http://elpa-aeo.mpcdf.mpg.de](http://elpa-aeo.mpcdf.mpg.de) for a full list of contributors
Electronic Structure Theory 101

INPUT

\( M \) three-dimensional vectors \( R_i \) defining the position of the individual nuclei \( i \) in space

\( M \) values \( Z_i \) for the charge defining the chemical species of the nuclei \( i \)
INPUT

$M$ three-dimensional vectors $\mathbf{R}_i$ defining the position of the individual nuclei $i$ in space

$M$ values $Z_i$ for the charge defining the chemical species of the nuclei $i$

OUTPUT

Spatial distribution of $N$ electrons, e.g., the electronic density $n(r)$ with

$$N = \int n(r) \, dr$$

Electronic Structure Theory 101

**INPUT**

- $M$ three-dimensional vectors $R_i$ defining the position of the individual nuclei $I$ in space.
- $M$ values $Z_I$ for the charge defining the chemical species of the nuclei $I$.

**OUTPUT**

- Spatial distribution of $N$ electrons, e.g., the electronic density $n(r)$ with

$$N = \int n(r) \, dr$$


**Eigenvalue Problem**

Computationally Dominant!
Electronic Structure Theory 101

Geometry & Species

Electronic Structure Theory

Electronic Structure

The Self-Consistency Cycle

Construct Matrix $A[n_i(r)]$ & $B$

Solve Eigenvalue Problem $A[n_i(r)] x = \lambda B x$

to obtain $n_{i+1}(r)$

Self-Consistency achieved?

Yes

No

$n_{i+1}(r)$ → $n_i(r)$
The Self-Consistency Cycle

1. Construct Matrix $A[n_i(r)]$ & $B$
2. Solve Eigenvalue Problem $A[n_i(r)] x = \lambda B x$ to obtain $n_{i+1}(r)$
3. Self-Consistency achieved?
   - Yes
   - No

A series of different eigenproblems with same $B$ are solved one after another.
Electronic Structure Theory 101

A series of different, but similar electronic structure theory problems are solved one after another.

- **Update Geometry**
- **Geometry & Species**
- **Electronic Structure Theory**
- **Compute Energy $E$ and Forces $F$ on Atoms**
- **⇒ Materials’ Properties**
The Potential-Energy Surface (PES) describes the dependence of the energy $E$ on the $M$ nuclear positions $R_i$.

The Potential-Energy Surface (PES) determines the dynamics of the nuclei.
Exploring the Potential-Energy Surface

The Potential-Energy Surface (PES) describes the dependence of the energy $E$ on the $M$ nuclear positions $R_i$.

The Potential-Energy Surface (PES) determines the dynamics of the nuclei.

- **Iterative Minimization ("Relaxation"):** Find the minima of the PES

  $\Rightarrow$ Most probable configuration(s), static properties, stabilities,…
Exploring the Potential-Energy Surface

The Potential-Energy Surface (PES) describes the dependence of the energy $E$ on the $M$ nuclear positions $R_i$.

The Potential-Energy Surface (PES) determines the dynamics of the nuclei.

- Iterative Minimization ("Relaxation"): Find the minima of the PES

- *Ab initio* Molecular Dynamics: Iteratively integrate the equations of motion

  ⇒ Quantitative explore the full Dynamics!
Exploring the Potential-Energy Surface

The Potential-Energy Surface (PES) determines the dynamics of the nuclei. The PES describes the dependence of the energy $E$ on the nuclear positions $R_i$.

- Iterative Minimization (“Relaxation”): Find the minima of the PES
- Transition-State Search: Find the minimum-energy path between minima of the PES
- Ab initio Molecular Dynamics: Iteratively integrate the equations of motion

⇒ Quantitatively explore the full dynamics!
Paracetamol exists in two different polymorphs: “Same paracetamol molecules, different crystalline order”
Paracetamol exists in two different polymorphs:
“Same paracetamol molecules, different crystalline order”

The two forms differ in their physico-chemical properties, e.g., solubility.
Example A: Paracetamol


Raman Spectra describing the response to electric fields typically used to differentiate between two polymorphs.
**Example A: Paracetamol**

*H. Shang, et al., New J. Physics, 20, 073040 (2018).*


(a) Form I  
(b) Form II

![Graphs comparing I(ω) for Form I and Form II with experimental and theoretical PBE+MBD/LDA data.](image-url)
Example B: Thermal-Barrier Coatings

Suppressing heat transport in thermal barrier coatings has driven the fuel-efficiency increase over the last 30 years.

THERMAL CONDUCTIVITY OF ZIRCONIA


Experiment: 2x2x2 Supercell, > 200ps AIMD / data point

Semi-empirical potential

ZrO₂ - PBEsol

Experiment

Semi-empirical MD:


THERMAL CONDUCTIVITY OF ZIRCONIA


---

Experiment:

Semi-empirical MD:

---

Graph showing the thermal conductivity of ZrO₂ as a function of temperature.

- 2x2x2 Supercell, > 200ps AIMD / data point
- Semi-empirical potential
- ZrO₂ - PBEsol
- Experiment

~600,000 EVP / data point
A series of different, but similar eigenvalue problems are solved one after another.

A series of different, but similar electronic structure theory problems are solved one after another.
Geometry & Species

Electronic Structure Theory

Electronic Structure

Compute Energy $E$ and Forces $F$ on Atoms

$\Rightarrow$ Materials’ Properties

A series of different, but similar eigenvalue problems are solved one after another.

A series of different, but similar electronic structure theory problems are solved one after another.

This Talk: How can this be exploited within ELPA to accelerate applications.
Overview:

Geometry Updates

Construct Matrix $B$

Guess for $n_0(r)$

Construct $A[n(r)]$

Solve Eigenvalue Problem

Self-Consistency achieved?

Compute other quantities of interest.

These are several self-consistency cycles.

What can we do across many electronic structure theory calculations?
Hypothesis: For similar problems, similar numerical settings and code paths should be most efficient.

Typical Approach: Manually test different settings or choose (possibly automatically) by educated guess.
Hypothesis: For similar problems, similar numerical settings and code paths should be most efficient.

Typical Approach: Manually test different settings or choose (possibly automatically) by educated guess.

ELPA’s Autotuning Solution: Explicitly test different (combination of) settings and then choose the optimal one. Possible settings include:

- Code Paths: ELPA 1 or ELPA 2 solver?
- Optimized Kernels: Generic, AVX2, or AVX512 kernel?
- Hybrid MPI/MP Parallelization: Number of threads / core
- GPU offload: Which routines should make use of GPUs?
Hypothesis: For similar problems, similar numerical settings and code paths should be most efficient.

Typical Approach: Manually test different settings or choose (possibly automatically) by educated guess.

ELPA’s Autotuning Solution: Explicitly test different (combination of) settings and then choose the optimal one. Possible settings include:

- **Code Paths:**
  - ELPA 1
  - or
  - ELPA 2
  - solver?

- **Optimized Kernels:**
  - Generic,
  - AVX2, or
  - AVX512
  - kernel?

- **Hybrid MPI/MP Parallelization:**
  - Number of threads / core

- **GPU offload:**
  - Which routines should make use of GPUs?

This requires ELPA to “remember” settings and timings from earlier calls across one run.
Necessary Technology:  
*API Redefinition*
Old ELPA API

![Code snippet from elsi_elpa.f90 in ELSI]

- Traditional **Scalapack-esque** interface
- **Stable and Here-to-Stay:**
  - All pre-2017 features are and will be accessible also in future
- This includes all internal accelerations and kernels.
- **New (optional) post-2017 features not accessible via the old API**
New ELPA API

- Object-oriented
- Get/Set API
- Mandatory and Optional Arguments are set separately
- Actual ELPA call has simple syntax
- ELPA Object survives iterations and cycles
New ELPA API

- Object-oriented
- Get/Set API
- Mandatory and Optional Arguments are set separately
- Actual ELPA call has simple syntax
- ELPA Object survives iterations and cycles

ELPA is able to store and exploit information across calls.
Example: Autotuning

- Usual setup
- Attach Autotuning
- Run your loops
- Set optimal settings once identified
Example: Autotuning

- Usual setup
- Attach Autotuning
- Run your loops
- Set optimal settings once identified

```plaintext
! Setup ELPA using the new API
success = elpa%setup()
!
! Autotuning setup
!tune_state => elpa%autotune_setup(ELPA_AUTOTUNE_MEDIUM, ELPA_AUTOTUNE_DOMAIN_REAL, error)
!
! Loop of ELPA calls
!do i=1, scf_cycles
! Autotuning still running?
!unfinished = elpa%autotune_step(tune_state)
!
! Use optimal setup once finished
!if (.not. (unfinished)) then
!call elpa%autotune_set_best(tune_state)
!endif
!
! ELPA call
!call elpaeigenvectors(a, ev, z, error)
!enddo
```
Example: Autotuning

- Usual setup
- Attach Autotuning
- Run your loops
- Set optimal settings once identified

Autotuning can be used during production iterations.

Manual Settings or Predefined Options
Hypothesis: For similar problems, similar numerical settings and code paths should be most efficient.

Typical Approach: Manually test different settings or choose (possibly automatically) by educated guess.

**ELPA’s Autotuning Solution:** Explicitly test different (combination of) settings and then choose the optimal one. Possible settings include:

- **Code Paths:**
  - ELPA 1
  - ELPA 2

- **Optimized Kernels:**
  - Generic, AVX2, or AVX512 kernel?

- **ELPA_AUTOTUNE_FAST**

- **Hybrid MPI/MP Parallelization:**
  - Number of threads / core

- **GPU offload:**
  - Which routines should make use of GPUs?

Reasonable domain-specific presets such as **ELPA_AUTOTUNE_FAST** available, more to come…
Autotuning with **ELPA_AUTOTUNE_FAST**

Accumulated Runtimes:

- **ZrO$_2$ – 96 atoms**
- **8 Intel Xeon E5-2698v3 CPUs**
  - (4 nodes, 32 cores/node)

*Accumulated run-times: Slight computational overhead observed in the first couple of iterations...*
Autotuning with ELPA_AUTOTUNE_FAST

**Accumulated Runtimes:**

- ZrO$_2$ – 96 atoms
- 8 Intel Xeon E5-2698v3 CPUs (4 nodes, 32 cores/node)

**Autotuning beneficial in the long-run**

- Autotuning can already be beneficial in a single-run!
- Generally, autotuning can eventually lead to a speed-up between 15-20% per SCF iteration.
- Autotuning speed-up retained across geometry changes (MD-ready!).

Slight computational overhead observed in the first couple of iterations...
Geometry Updates

These are several self-consistency cycles.

⇒ (A) Exploit ELPA-Autotuning for a series of calculations

Overview:

Construct Matrix $B$

Guess for $n_0(r)$

Construct $A[n_i(r)]$

Solve Eigenvalue Problem

Self-Consistency achieved?

Compute other quantities of interest.
Overview:

Geometry Updates

- Construct Matrix $B$
- Guess for $n_0(r)$
- Construct $A[n_i(r)]$
- Solve Eigenvalue Problem
- Self-Consistency achieved?
- Compute other quantities of interest.

These are several self-consistency cycles.

⇒ (A) Exploit ELPA-Autotuning for a series of calculations

This is one self-consistency cycle. (Several Eigenvalue Problems)
The Self-Consistency Cycle

Construct Matrix $B$

Educated guess for $n_0(r)$

Construct $A[n_i(r)]$

Solve Eigenvalue Problem to obtain $n_{i+1}(r)$

Self-Consistency achieved?

Yes

Compute other quantities of interest.

No

Some Remarks:

- Typical number of iterations: 10-100
- Convergence accelerated by mixing schemes
- Convergence/Self-consistency monitored via:
  - Change in density $n_i(r)$
  - Change in energy $E_i(r)$
  - Change in forces $F_i$
  - Change in eigenenergies $\epsilon_i$
The Self-Consistency Cycle

1. Construct Matrix $B$
2. Educated guess for $n_0(r)$
3. Construct $A[n_i(r)]$
4. Solve Eigenvalue Problem to obtain $n_{i+1}(r)$
5. Self-Consistency achieved?
   - Yes
   - No
6. Compute other quantities of interest.

Some Remarks:

- Typical number of iterations: 10-100
- Convergence accelerated by mixing schemes
- Convergence/Self-consistency monitored via:
  - Change in density $n_i(r)$
  - Change in energy $E_i(r)$
  - Change in forces $F_i$
  - Change in eigenenergies $\varepsilon_i$

Extensive & Variational
Intensive & Non-Variational
The Self-Consistency Cycle

Educated guess for \( n_0(r) \)

Construct \( A[n_i(r)] \)

Solve Eigenvalue Problem to obtain \( n_{i+1}(r) \)

Self-Consistency achieved?

\( n_{i+1}(r) \rightarrow n_i(r) \)

No

Construct Matrix \( B \)

Compute other quantities of interest.

Yes

Some Remarks:

- Typical number of iterations: 10-100
- Convergence accelerated by mixing schemes
- Convergence/Self-consistency monitored via:
  - Change in density \( n_i(r) \)
  - Change in energy \( E_i(r) \)
  - Change in forces \( F_i \)
  - Change in eigenenergies \( \epsilon_i \)

Extensive & Variational
Intensive & Non-Variational

IDEA:
Use mixed precision to reduce computational cost
Perform the first self-consistency cycles with cheaper single precision routines.
Mixed Precision Calculations

- **Mixed Precision Calculations**
- **Full SCF iteration**
- **ELPA**
- **Standard double precision (DP)**
- **Single Precision (SP)**
- **SP routines ~20% faster**
- **Eigenvalue problem dominates comp. cost for larger systems.**

- **ZrO$_2$ – 6-768 atoms**
- **8 Intel Xeon E5-2698v3 CPUs (4 nodes, 32 cores/node)**

![Graph](image)
Mixed Precision Calculations

ZrO$_2$ – 6-768 atoms
8 Intel Xeon E5-2698v3 CPUs
(4 nodes, 32 cores/node)

- System size needed for a relaxation ("PES minima search").
- System size needed to model atomic defects in the crystal.
- System size needed for a thermal conductivity calculation in pristine ZrO$_2$.

- SP routines ~20% faster
- Eigenvalue problem dominates comp. cost for larger systems.

**Run time [sec]**

- Full SCF iteration
- ELPA

**Standard double precision (DP)**

**Single Precision (SP)**
How far can we go?

Extensive: Change in density $n_i(r)$

Monitoring the convergence of the self-consistency cycle

- Single-Precision routines do not allow to reach convergence.
- “First” few iterations can be performed with single-precision without loss of accuracy.
- Break-down point of single-precision routines system-size-dependent.
How far can we go?

**Extensive:** Change in density $n_i(r)$

**Intensive:** Change in eigenenergies $\epsilon_i$

- Single-Precision routines do not allow to reach convergence.
- “First” few iterations can be performed with single-precision without loss of accuracy.
- Break-down point of single-precision routines system-size-dependent.

⇒ Intensive Quantities naturally less affected by system-size dependence.
How far can we go?

- Extensive: Change in density $n_i(r)$
- Intensive: Change in eigenenergies $\epsilon_i$

CAVEAT: Actual breakdown points are not only system-size, but also system dependent!

- Single-Precision routines do not allow to reach convergence.
- “First” few iterations can be performed with single-precision without loss of accuracy.
- Break-down point of single-precision routines system-size-dependent.

⇒ Intensive Quantities naturally less affected by system-size dependence.
These are several self-consistency cycles.

⇒ (A) Exploit ELPA-Autotuning for a series of calculations

This is one self-consistency cycle.
(Several Eigenvalue Problems)
⇒ (B) Exploit Single/Double Precision routines in ELPA-AEO for a series of Eigenvalue Problem

Overview:

Geometry Updates

Construct Matrix $B$

Guess for $n_0(r)$

Construct $A[n_i(r)]$

Solve Eigenvalue Problem

Self-Consistency achieved?

Compute other quantities of interest.
Overview:

**Geometry Updates**

- Construct Matrix $B$
- Guess for $n_0(r)$
- Construct $A[n_i(r)]$
- Solve Eigenvalue Problem
- Self-Consistency achieved?
- Compute other quantities of interest.

These are several self-consistency cycles.

⇒ (A) Exploit ELPA-Autotuning for a series of calculations

This is one self-consistency cycle.
(Several Eigenvalue Problems)

⇒ (B) Exploit Single/Double Precision routines in ELPA-AEO for a series of Eigenvalue Problem

This is one self-consistency iteration.
(One Eigenvalue Problem)
Accelerating One Iteration

**Ivy Bridge**: Hydra@MPCDF – 20 cores/node  
**Skylake**: Cobra@MPCDF – 40 cores/node

![Graph showing the time per SCF iteration for Ivy Bridge and Skylake with and without AVX512 optimization.](image)

- Ivy Bridge: Hydra@MPCDF – 20 cores/node  
- Skylake: Cobra@MPCDF – 40 cores/node

Optimizations in ELPA directly translate into an application speed-up.

**TiO$_2$ slab/surface with 3,500 atoms and 100 Å vacuum**
SUMMARY:

Geometry Updates

- Construct Matrix $B$
- Guess for $n_0(r)$
- Construct $A[n_i(r)]$
- Solve Eigenvalue Problem
- Self-Consistency achieved?
- Compute other quantities of interest.

These are several self-consistency cycles.

⇒ (A) Exploit ELPA-Autotuning for a series of calculations

This is one self-consistency cycle.
(Several Eigenvalue Problems)

⇒ (B) Exploit Single/Double Precision routines in ELPA-AEO for a series of Eigenvalue Problem

This is one self-consistency iteration.
(One Eigenvalue Problem)

⇒ (C) Exploit optimizations in ELPA-AEO for single Eigenvalue Problem
Acknowledgments

Danilo Simoes Brambila FHI

(now at some fancy Startup)

Matthias Scheffler FHI
Acknowledgments

Danilo Simoes Brambila
FHI

(now at some fancy Startup)

Matthias Scheffler
FHI

The ELPA-AEO Consortium

http://elpa-aeo.mpcdf.mpg.de
Acknowledgments

Danilo Simoes Brambila
FHI

(now at some fancy Startup)

Matthias Scheffler
FHI

The ELPA-AEO Consortium

http://elpa-aeo.mpcdf.mpg.de

Volker Blum
Victor Yu
Acknowledgments

Danilo Simoes Brambila FHI
(now at some fancy Startup)

Matthias Scheffler FHI

Volker Blum
Victor Yu

All features discussed in this talk, including the new API are available and usable in ELSI.

The ELPA-AEO Consortium
http://elpa-aeo.mpcdf.mpg.de
This is one self-consistency iteration. (One Eigenvalue Problem)

⇒ (C) Exploit optimizations in ELPA-AEO for single Eigenvalue Problem

These are several self-consistency cycles.

⇒ (A) Exploit ELPA-Autotuning for a series of calculations

⇒ (B) Exploit Single/Double Precision routines in ELPA-AEO for a series of Eigenvalue Problem

SUMMARY:

Construct Matrix $B$

Guess for $n_0(r)$

Construct $A[n_i(r)]$

Solve Eigenvalue Problem

Self-Consistency achieved?

Compute other quantities of interest.