MolSSI Workshop – ELSI Conference:
"Solving or Circumventing Eigenvalue Problems in Electronic Structure Theory"
Richmond, Virginia – 26th August 2018

Recent Advancements in ELPA: Best Practices in Real Applications

Danilo Simoes Brambila and Christian Carbogno

MAX-PLANCK-GESELLSCHAFT

Fritz-Haber-Institut
der Max-Planck-Gesellschaft,
Berlin - Germany

ELPA-AEO is an *Application-Driven* Team-Effort

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

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

Fritz Haber

ELPA-AEO is an *Application-Driven* Team-Effort

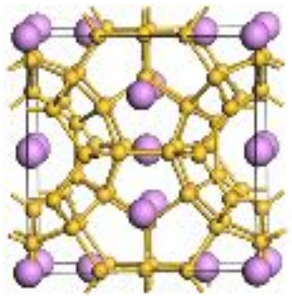
See http://elpa-aeo.mpcdf.mpg.de for a full list of contributors

The ab initio molecular simulations package

FHI aims

Electronic Structure Theory
Density Functional Theory

Solid State Physics & Material Science

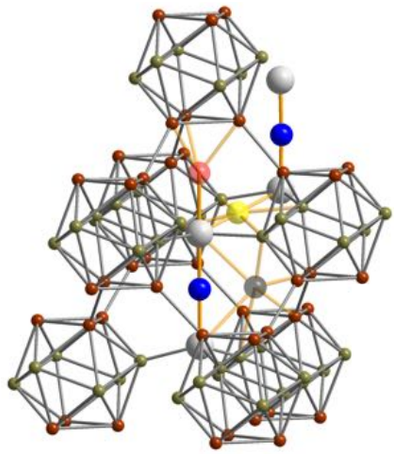


Thermoelectric Clathrates

A. Bhattacharya, *et al.*,
Phys. Rev. Lett. **118**, 236401 (2017).

Thermoelectric Boron Carbides

K. Rasim, *et al.*,
Ang. Chemie **57**, 6130 (2018).



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 for a full list of contributors

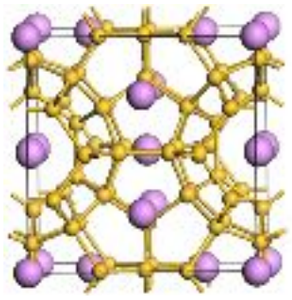
Electronic Structure Theory
Density Functional Theory

Solid State Physics & Material Science

Quantum Chemistry

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

Fritz Haber Institute

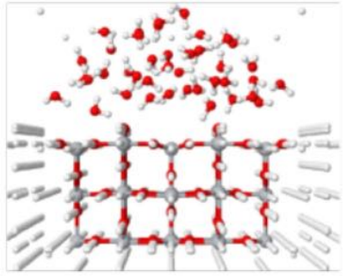


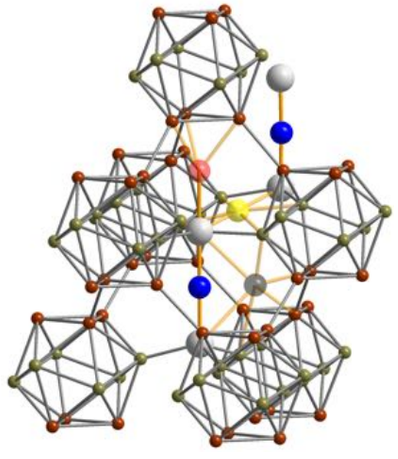
Thermoelectric Clathrates

A. Bhattacharya, *et al.*,
Phys. Rev. Lett. **118**, 236401 (2017).

Free Energy Barriers for H_2O on TiO_2

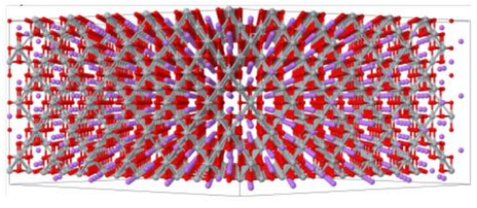
T. Stecher, *et al.*,
Phys. Rev. Lett. **117**, 276001 (2017).





Functional Boron Carbides

K. Rasim, *et al.*,
Ang. Chemie **57**, 6130 (2018).



Occupations Disorder in $Li_4Ti_5O_{12}$ Batteries

H. Heenen *et al.*,
Nano Letters **17,** 3884 (2017).

Christoph Scheurer

ELPA-AEO is an *Application-Driven* Team-Effort

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

Algorithmic Developments & Advancements

BERGISCHE UNIVERSITÄT WUPPERTAL

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

Fritz Haber Institute

Bruno Lang

ELPA-AEO is an *Application-Driven* Team-Effort

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

Algorithmic Developments & Advancements

BERGISCHE UNIVERSITÄT WUPPERTAL

Massive Parallel Numerical Implementation

INFORMATICS V

TUM

Architecture-specific Optimization, Adaption, and Code Maintenance

MPCDF Max Planck Computing & Data Facility

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

Fritz Haber Institute

TUM

ELPA-AEO is an *Application-Driven* Team-Effort

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

Algorithmic Developments & Advancements

BERGISCHE UNIVERSITÄT WUPPERTAL

Massive Parallel Numerical Implementation

INFORMATICS V

TUM

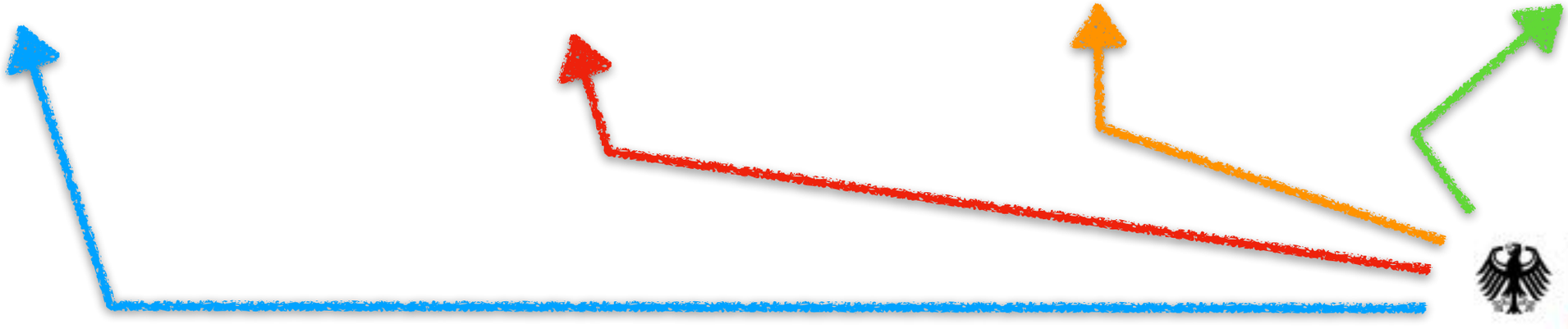
Architecture-specific Optimization, Adaption, and Code Maintenance

MPCDF Max Planck Computing & Data Facility

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

Fritz Haber Institute

TUM



GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

ELPA-AEO is an *Application-Driven* Team-Effort

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

Algorithmic Developments & Advancements

Massive Parallel Numerical Implementation

Architecture-specific Optimization, Adaption, and Code Maintenance

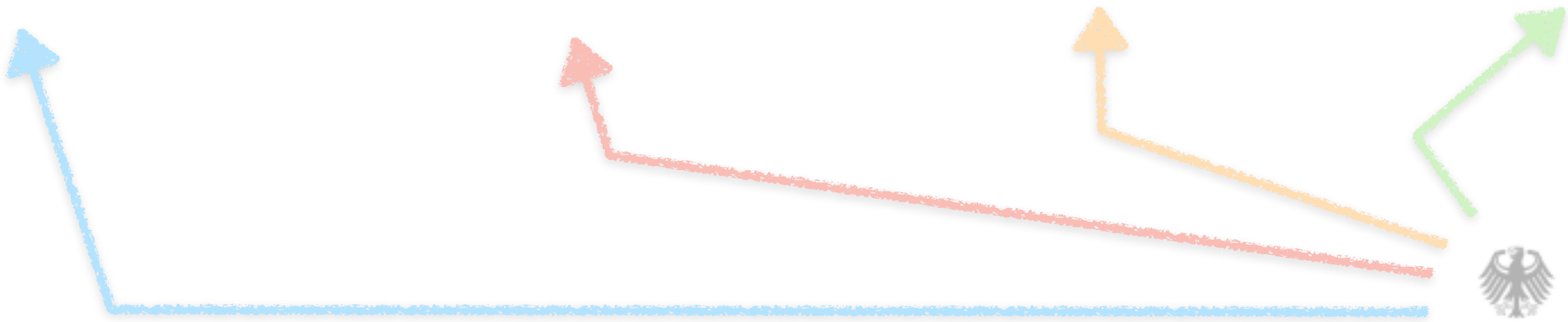
Code Testing, Benchmarking, and Validation in *Real Applications*

BERGISCHE UNIVERSITÄT WUPPERTAL

INFORMATICS V

MPCDF Max Planck Computing & Data Facility

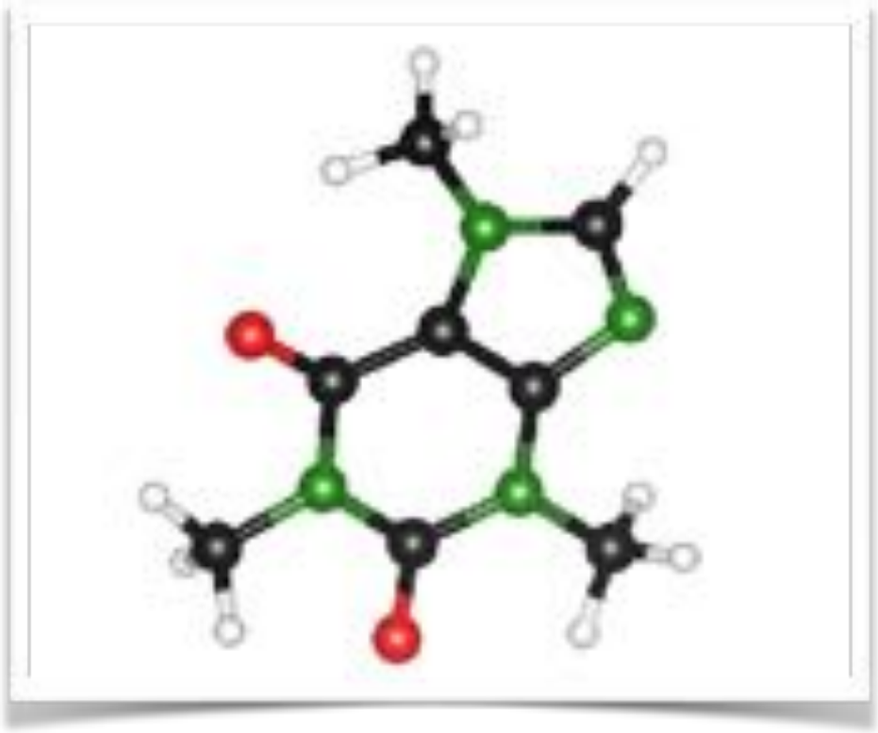
Fritz Haber Institute



GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

Electronic Structure Theory 101

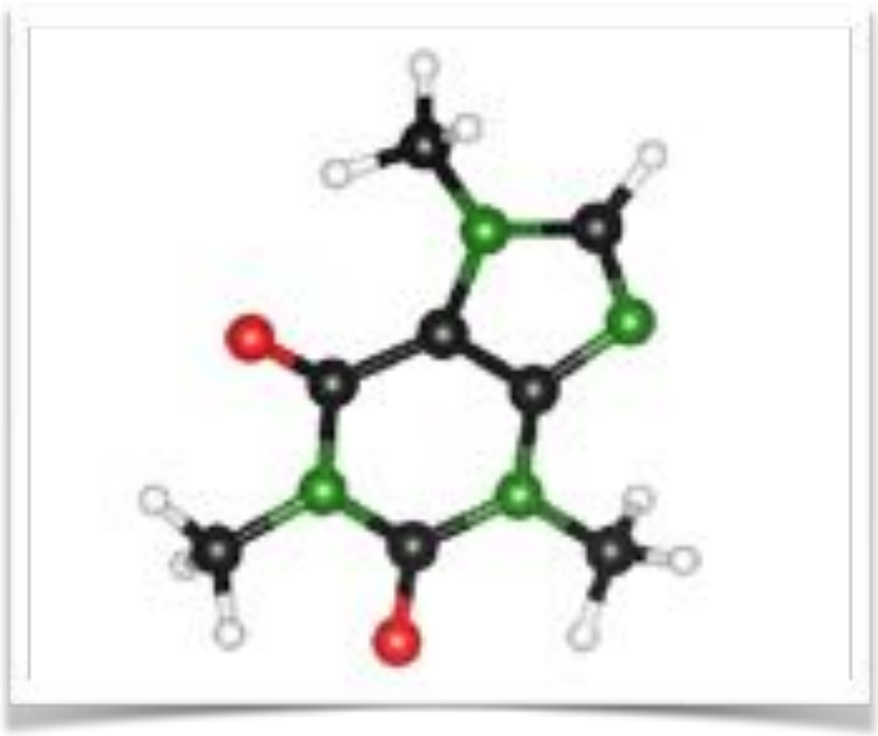


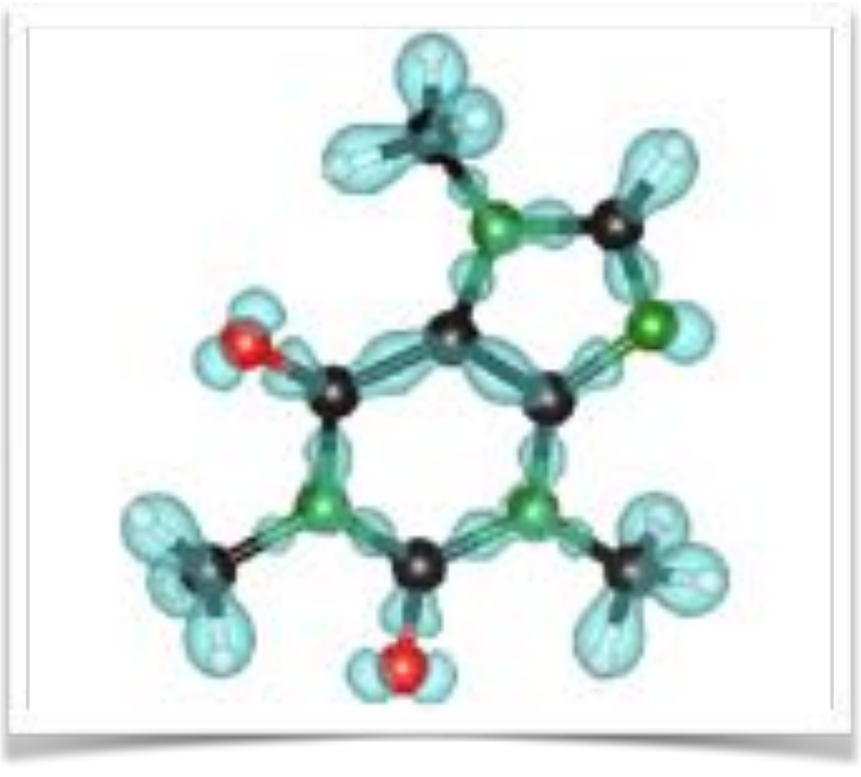
INPUT

M three-dimensional vectors $\boldsymbol{R_I}$ 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*

Electronic Structure Theory 101





INPUT

M three-dimensional vectors $\boldsymbol{R}_I$ 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*

Electronic Structure Theory

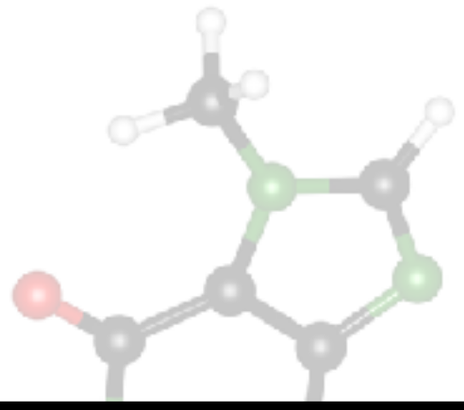
OUTPUT

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

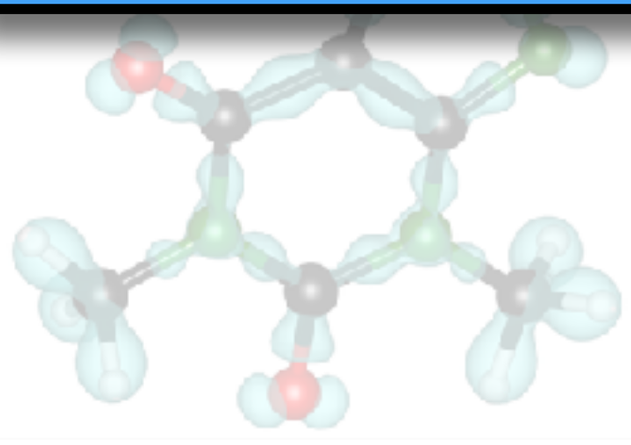
$$N = \int n(\mathbf{r}) d\mathbf{r}$$

in ***density-functional theory*** (1998 Nobel Prize).

Electronic Structure Theory 101



Eigenvalue Problem Computationally Dominant!



INPUT

M three-dimensional vectors $\boldsymbol{R}_I$ 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*

Electronic Structure Theory

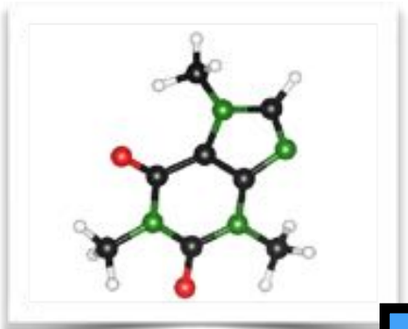
OUTPUT

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

$$N = \int n(\mathbf{r}) d\mathbf{r}$$

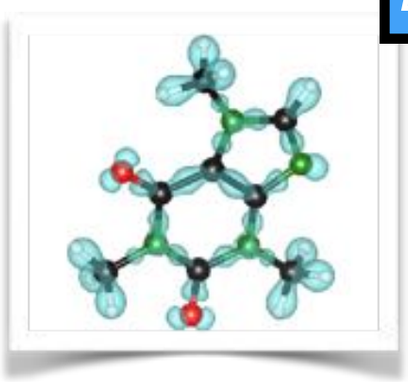
in ***density-functional theory*** (1998 Nobel Prize).

Electronic Structure Theory 101



Geometry & **Species**

Electronic Structure Theory



Electronic Structure

The Self-Concistency Cycle

Construct Matrix $\underline{\boldsymbol{A}}[n_i(\boldsymbol{r})]$ & $\underline{\boldsymbol{B}}$

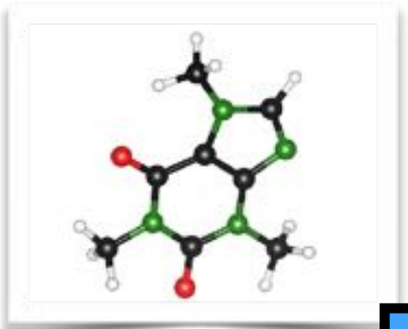
Solve **Eigenvalue Problem**
$\underline{\boldsymbol{A}}[n_i(\boldsymbol{r})]\ \boldsymbol{x} = \lambda\ \underline{\boldsymbol{B}}\ \boldsymbol{x}$
to obtain $n_{i+1}(\boldsymbol{r})$

Self-Consistency achieved?

No $n_{i+1}(r) \rightarrow n_i(r)$

Yes

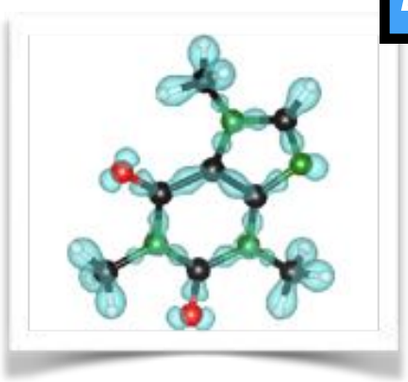
Electronic Structure Theory 101



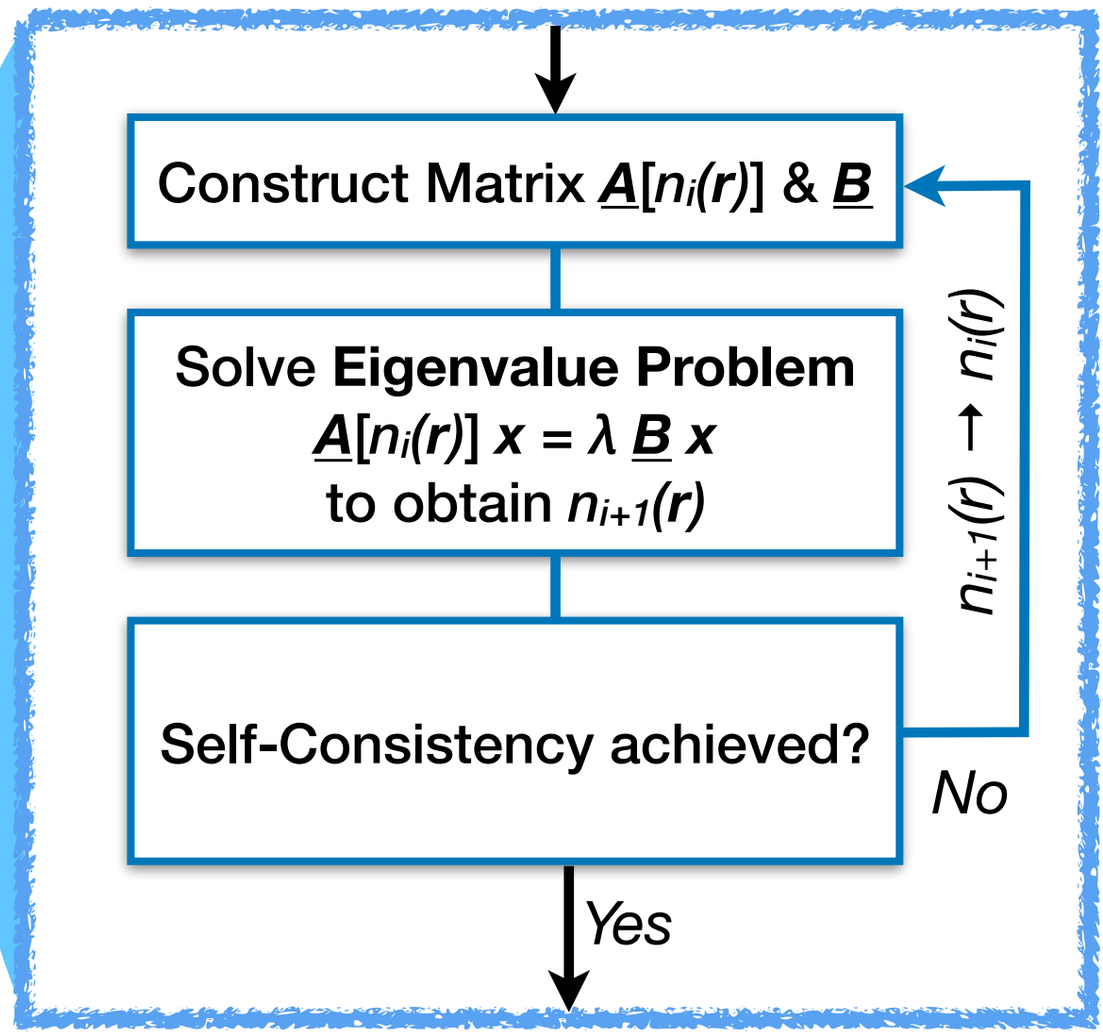
Geometry & **Species**

Electronic Structure Theory

Electronic Structure



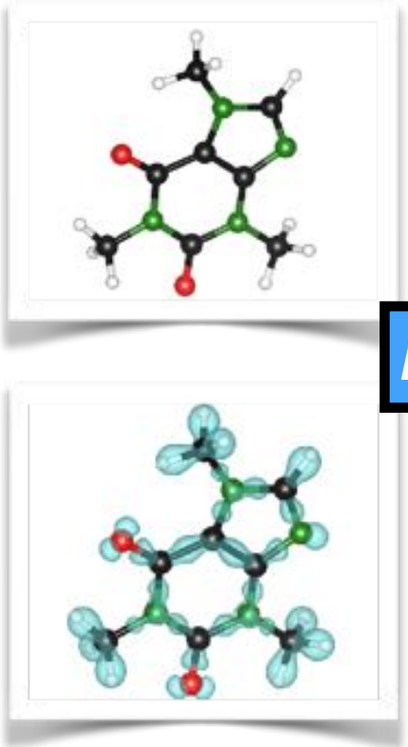
The Self-Concistency Cycle



A series of different eigenproblems with *same* $\underline{B}$ are solved one after another.

Electronic Structure Theory 101

Update Geometry



Geometry & Species

Electronic Structure Theory

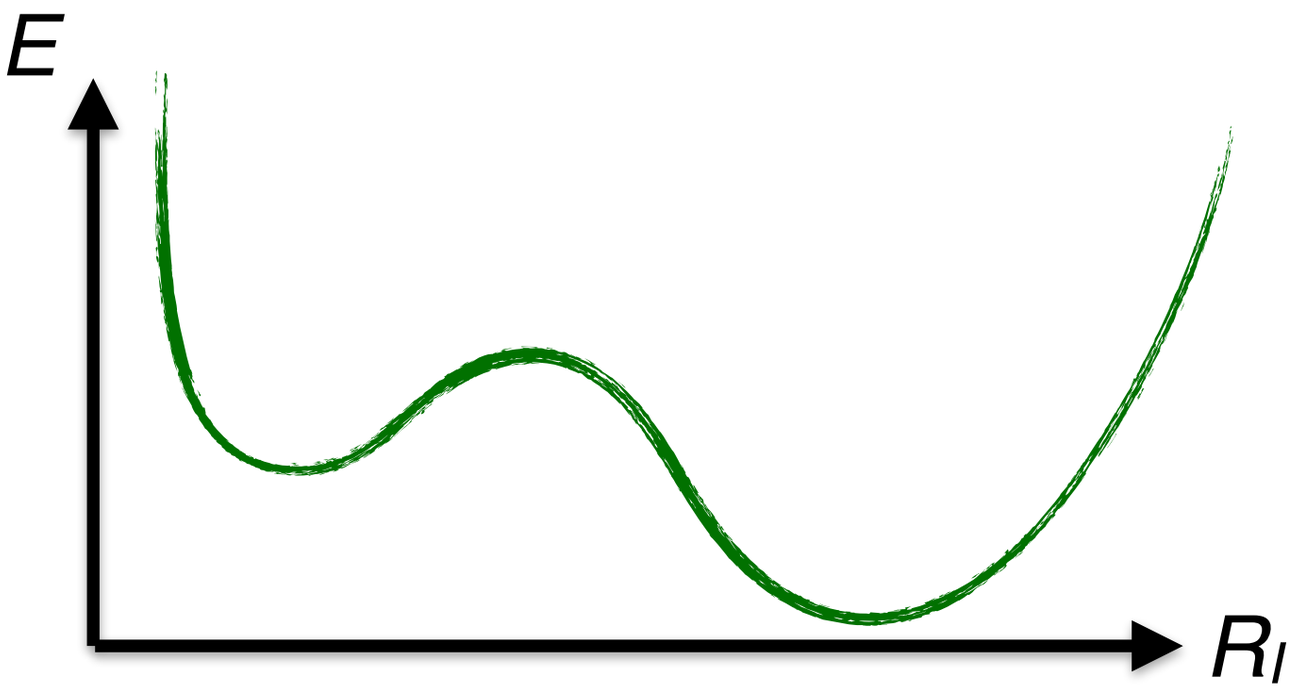
Electronic Structure

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

⇒ Materials' Properties

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

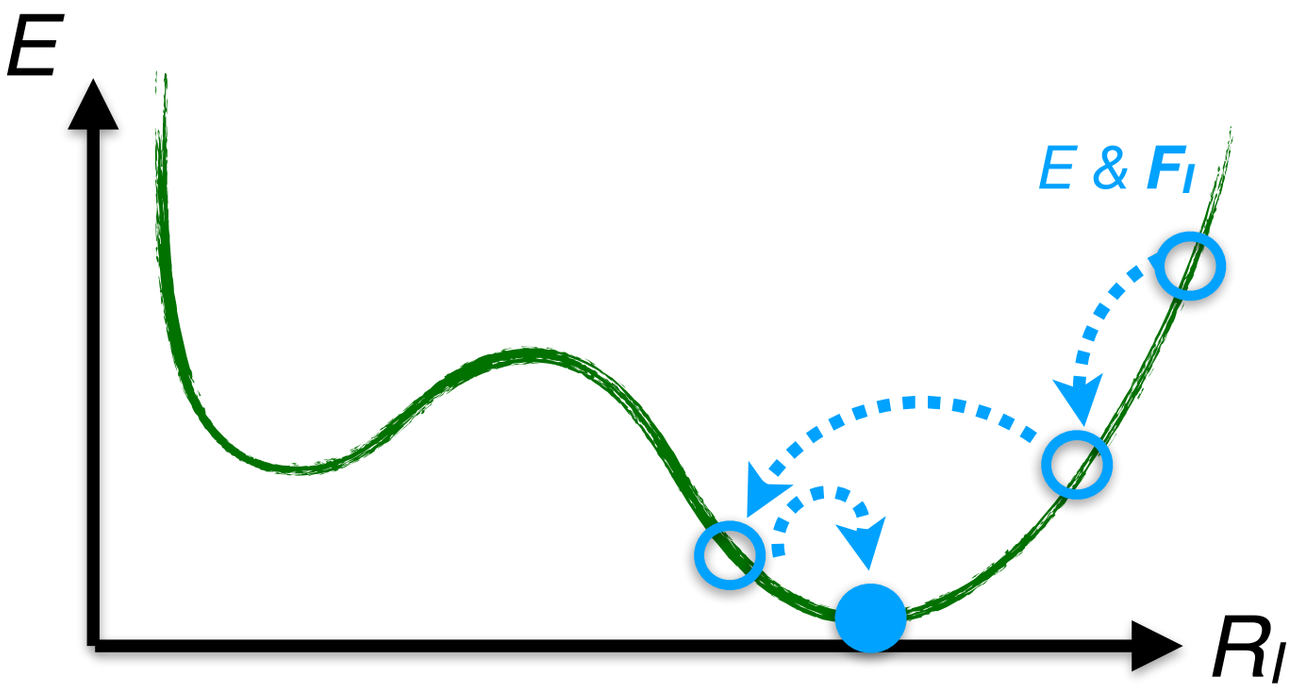
Exploring the Potential-Energy Surface



The *Potential-Energy Surface (PES)* describes the **dependence** of the **energy** *E* on the *M* **nuclear positions** $\boldsymbol{R_I}$.

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

Exploring the Potential-Energy Surface



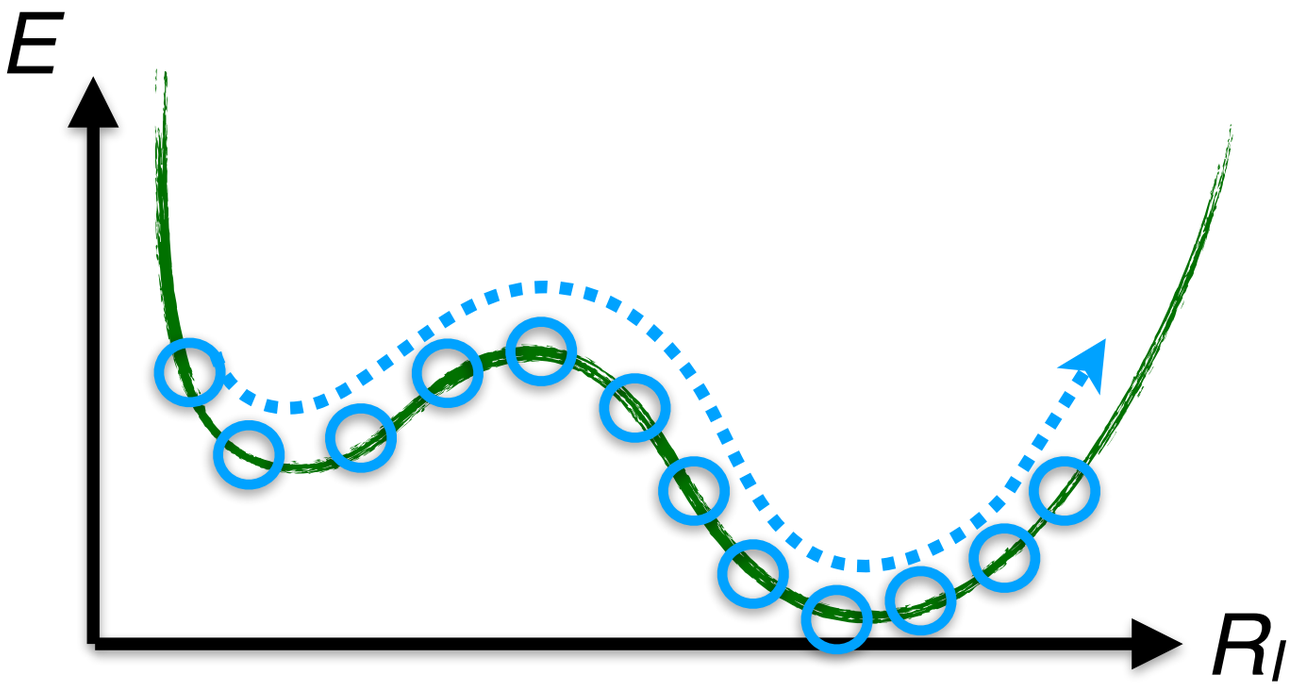
The *Potential-Energy Surface (PES)* describes the **dependence** of the **energy** *E* on the *M* **nuclear positions** $\boldsymbol{R_I}$.

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

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

  ⇒ 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** $\boldsymbol{R_I}$.

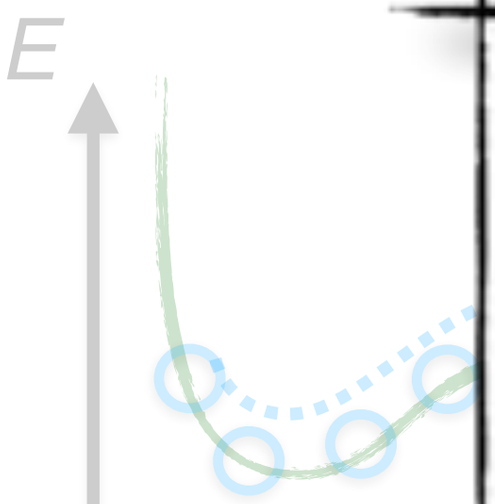
The *Potential-Energy Surface (PES)* determines the **dynamics** of the **nuclei.**

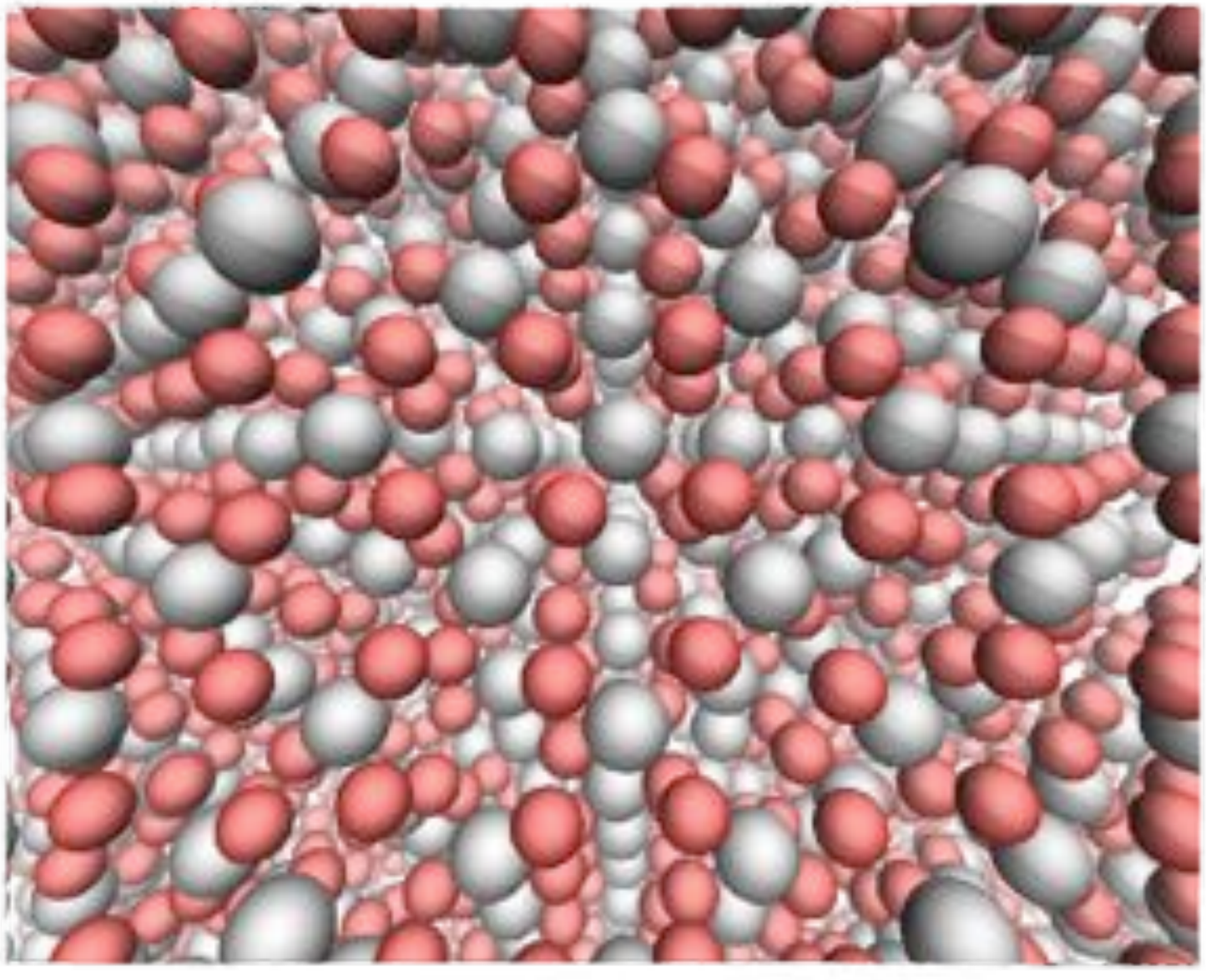
- **Iterative Minimization ("Relaxation"):** Find the *minima* of the PES

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

  ⇒ **Quantitative explore the full Dynamics!**

Exploring the Potential-Energy Surface





Surface (PES)

ndence of the

lear positions R_I.

Surface (PES)

ics of the **nuclei.**

- **Iterative Mini**
- **Transition-Sta** **inima of the PES**
- ***Ab initio* Mole** **otion**

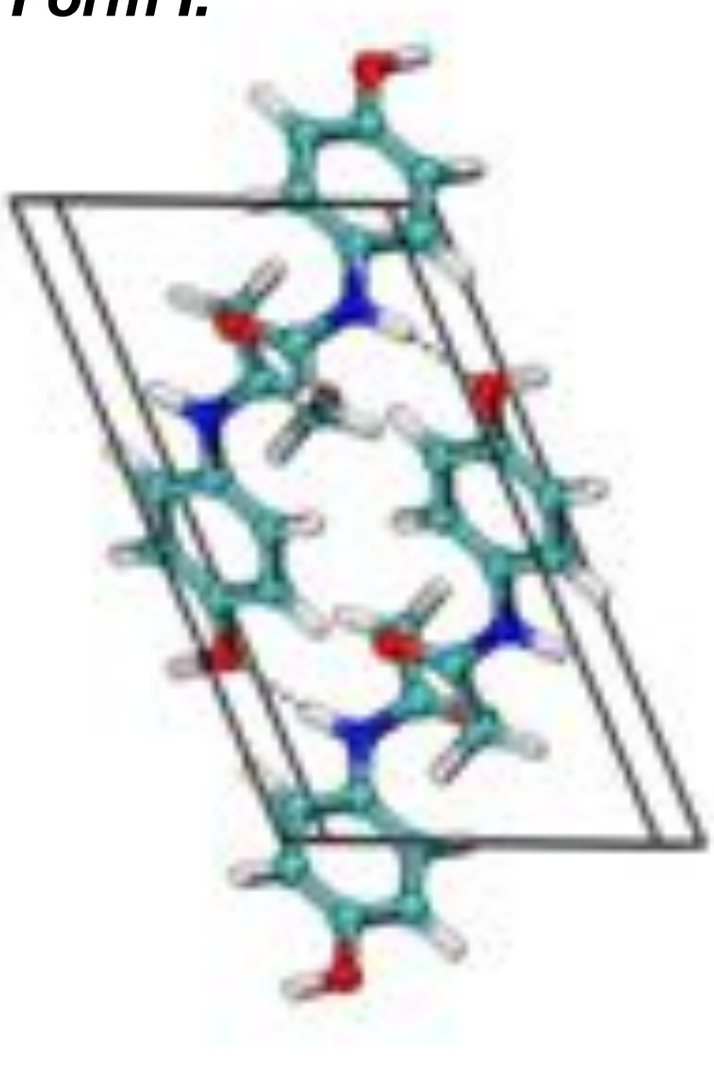
⇒ **Quantitative explore the full Dynamics!**

Example A: Paracetamol

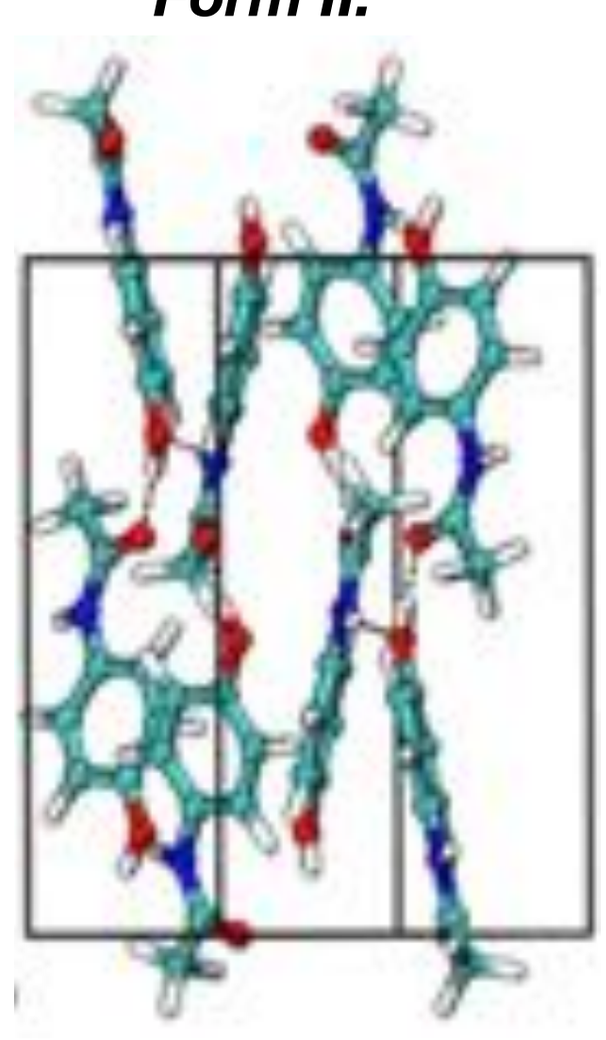
H. Shang, et al., New J. Phys., ***DOI:****10.1088/1367-2630/aace6d*

Paracetamol exists in **two** different **polymorphs:**
"Same paracetamol molecules, different crystalline order"

Form I:



Form II:

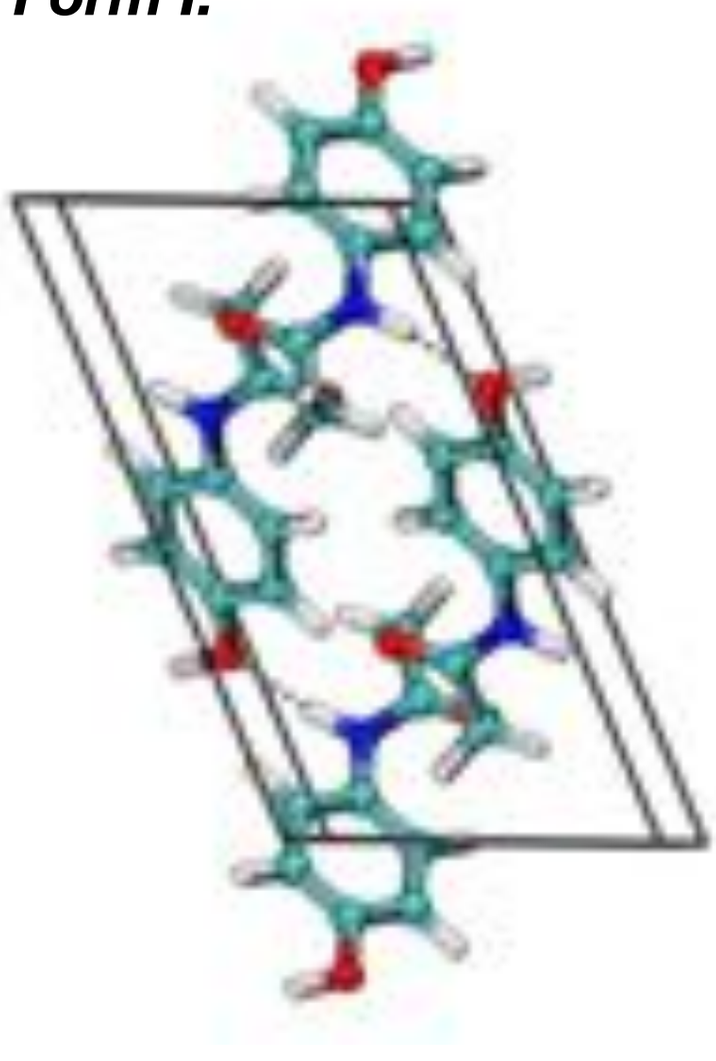


Example A: Paracetamol

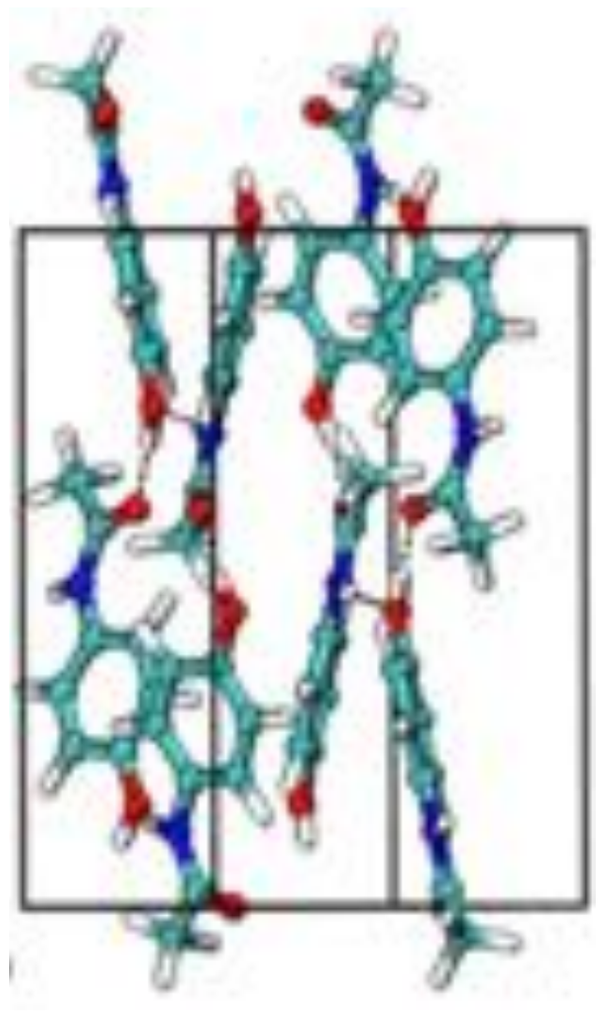
H. Shang, et al., New J. Phys., ***DOI:****10.1088/1367-2630/aace6d*

Paracetamol exists in two different polymorphs:
"Same paracetamol molecules, different crystalline order"

Form I:



Form II:

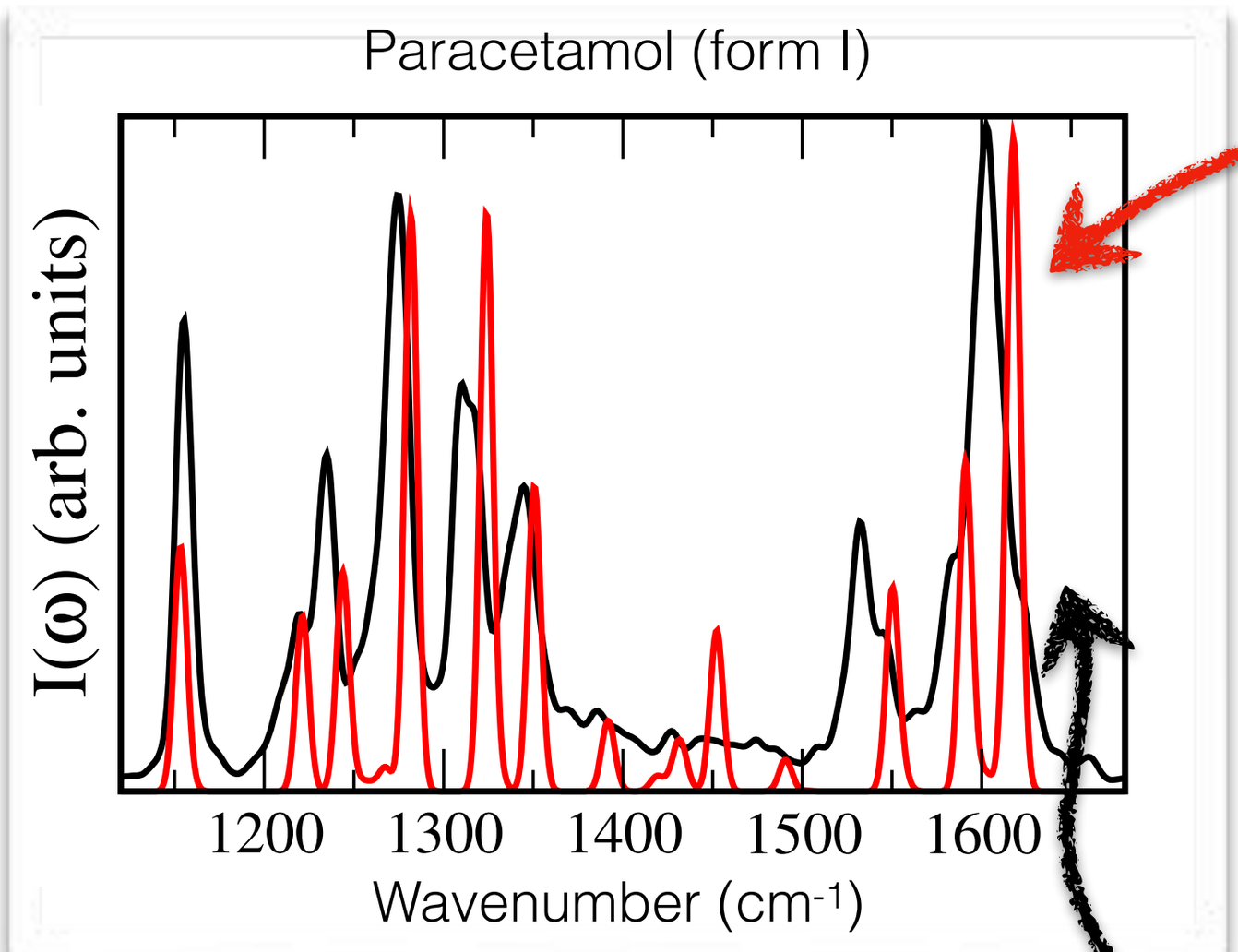


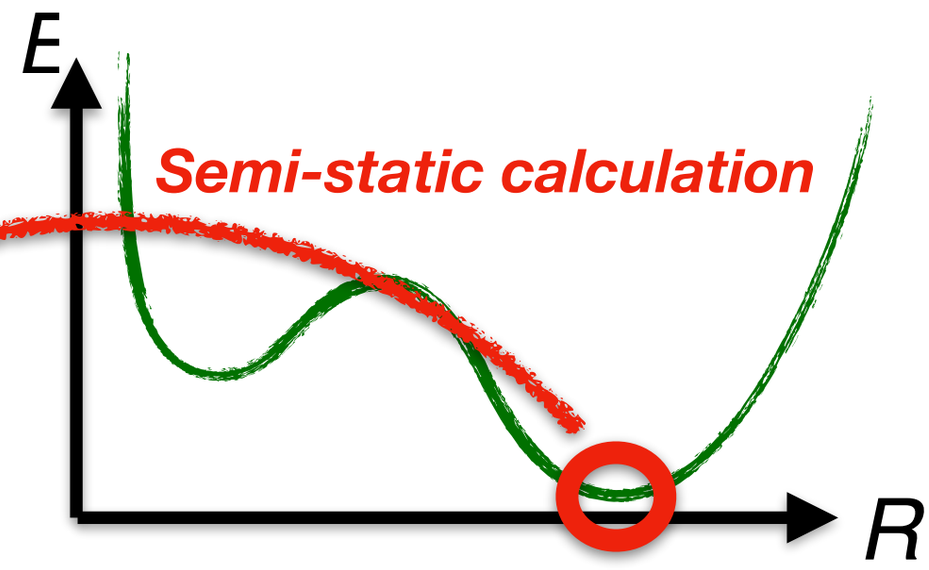
The two forms differ in their physo-chemical properties, e.g., *solubility.*

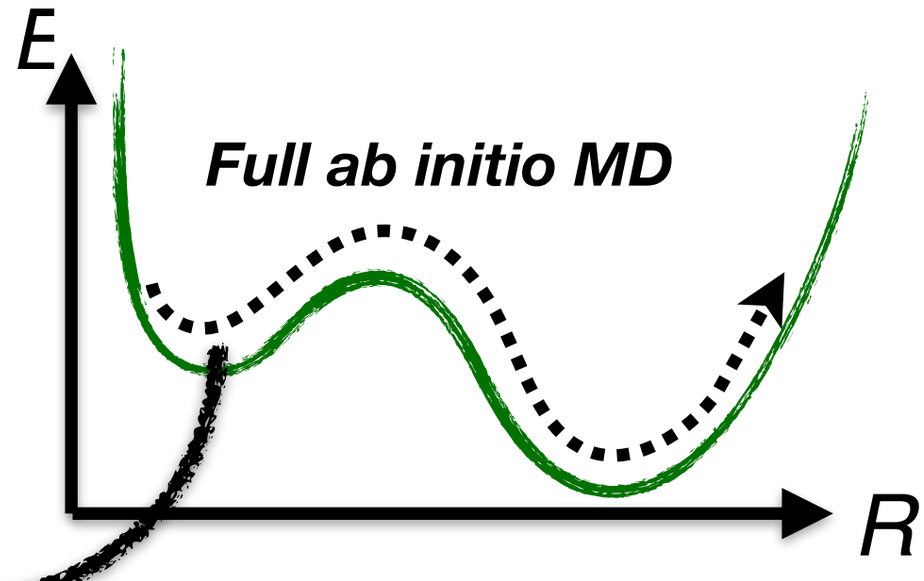
Example A: Paracetamol

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

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



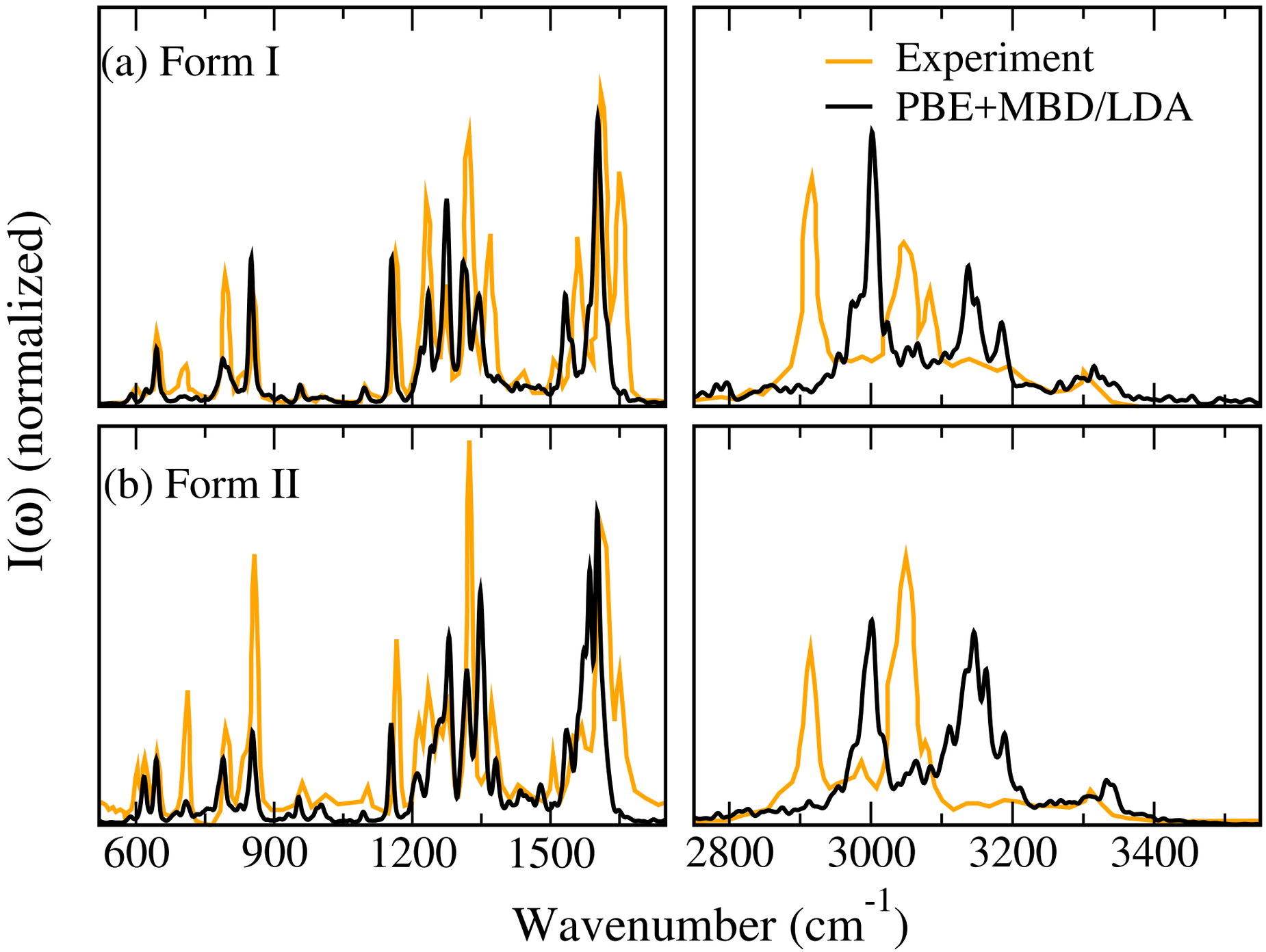


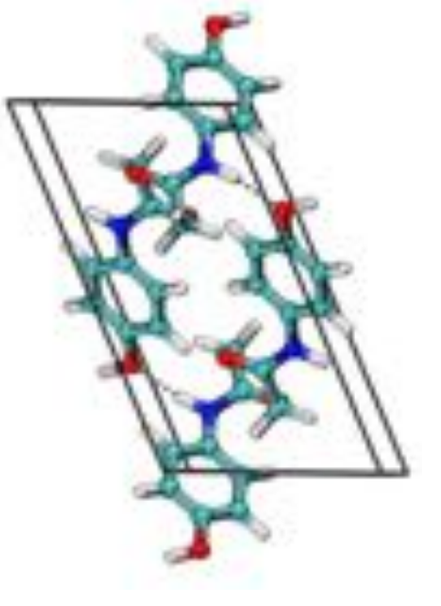


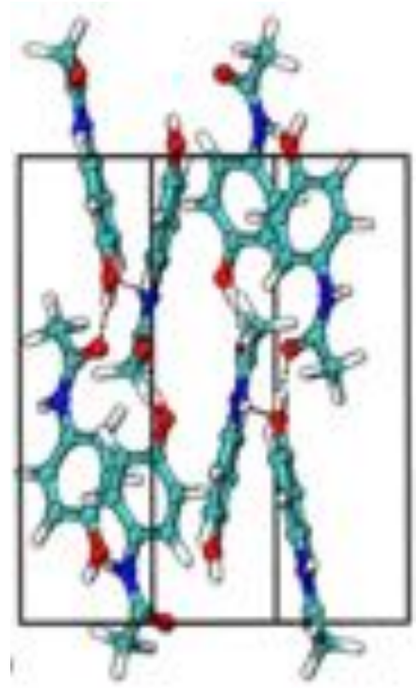
Example A: Paracetamol

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

Exp.: J. B. Nanubolu and J. C. Burley. *Mol. Pharm.*, **9**,1544 (2012).

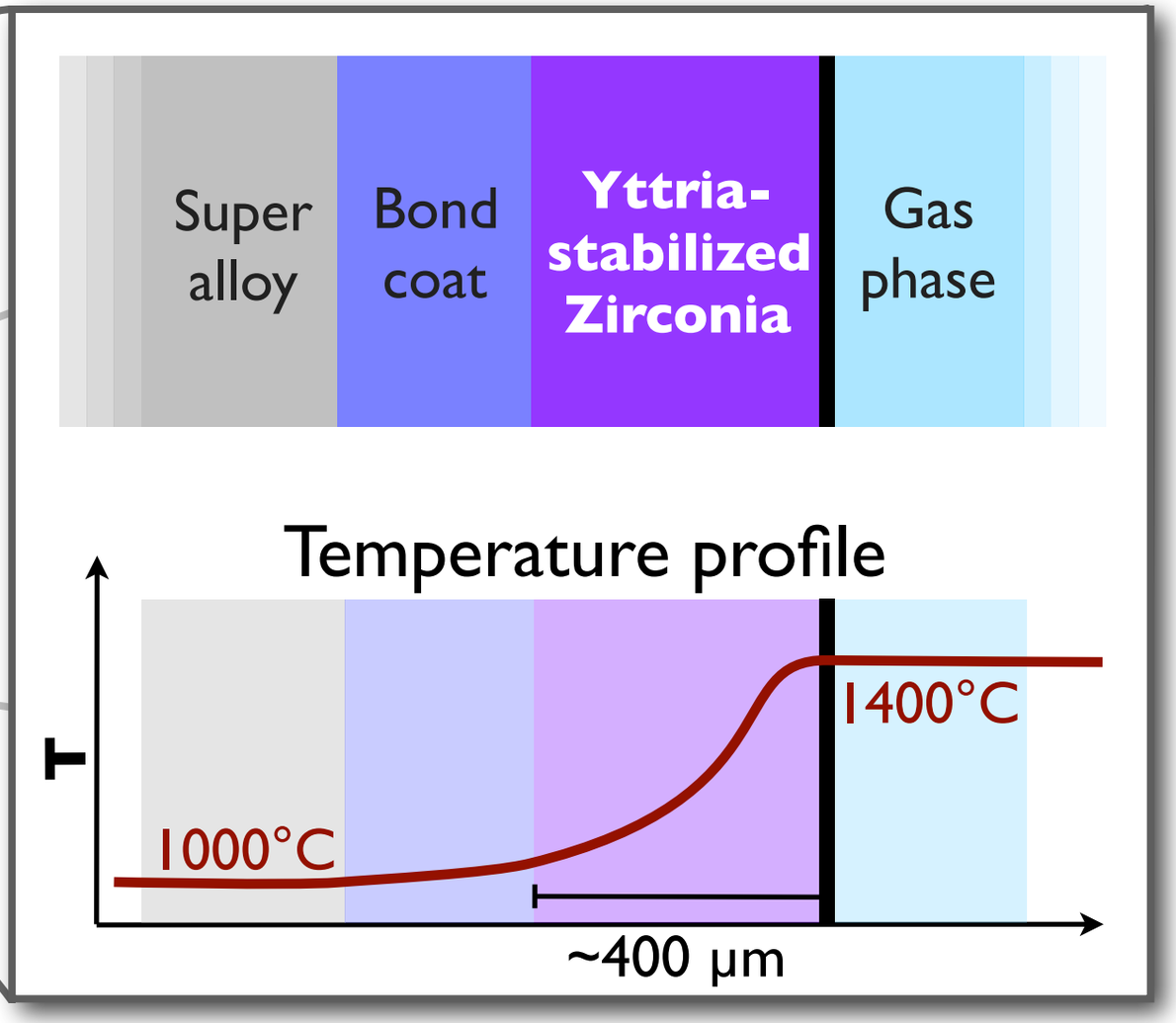






Example B: Thermal-Barrier Coatings

CFM 56-7 airplane engine

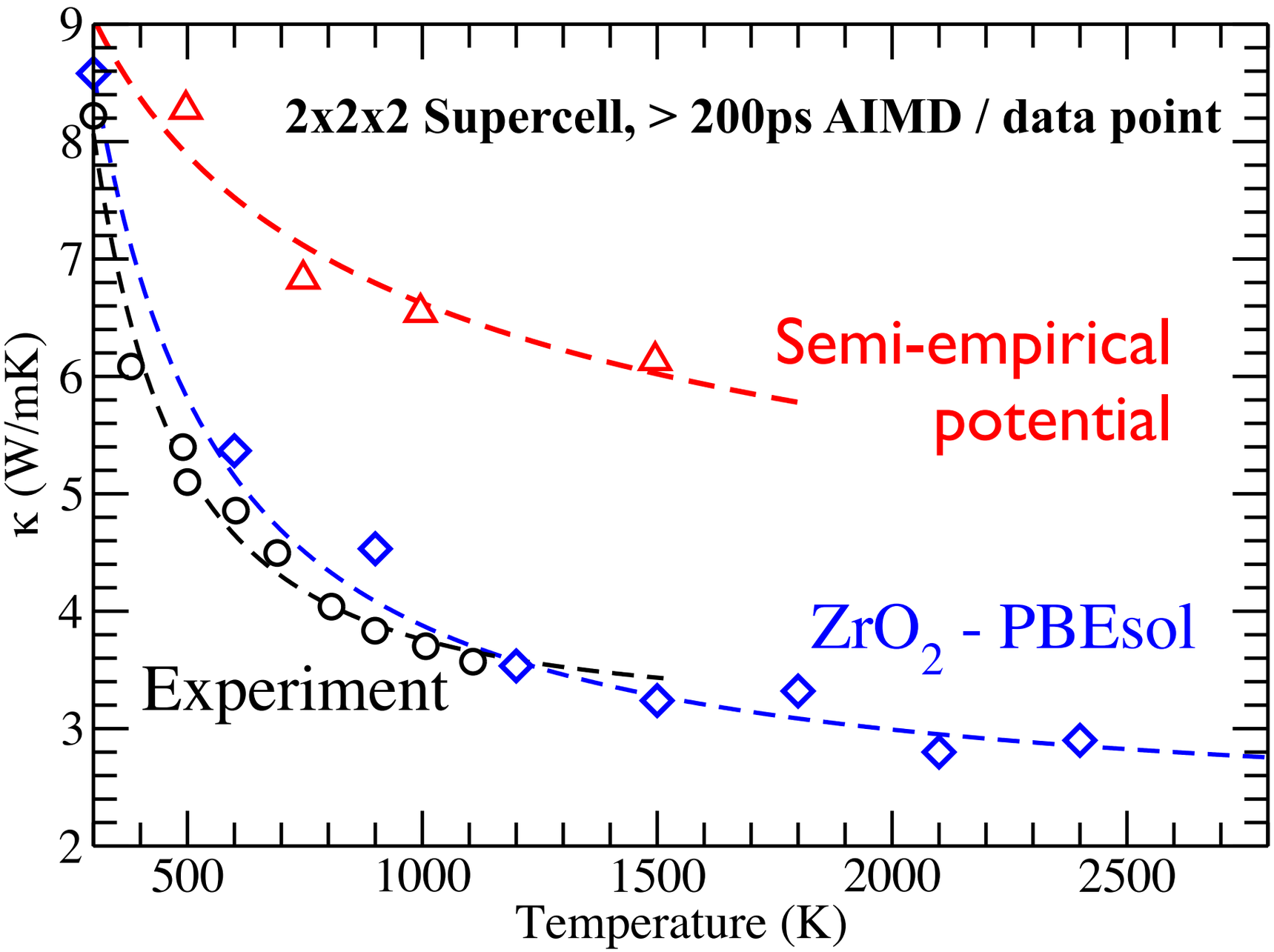


Suppressing heat transport in **thermal barrier coatings** has driven the fuel-efficiency increase over the last 30 years.

D. R. Clarke & C. G. Levi, *Ann. Rev. Mat. Res.*, **33**, 383 (2003).

THERMAL CONDUCTIVITY OF ZIRCONIA

C. Carbogno, R. Ramprasad, and M. Scheffler, *Phys. Rev. Lett.* **118**, 175901 (2017).



Experiment:

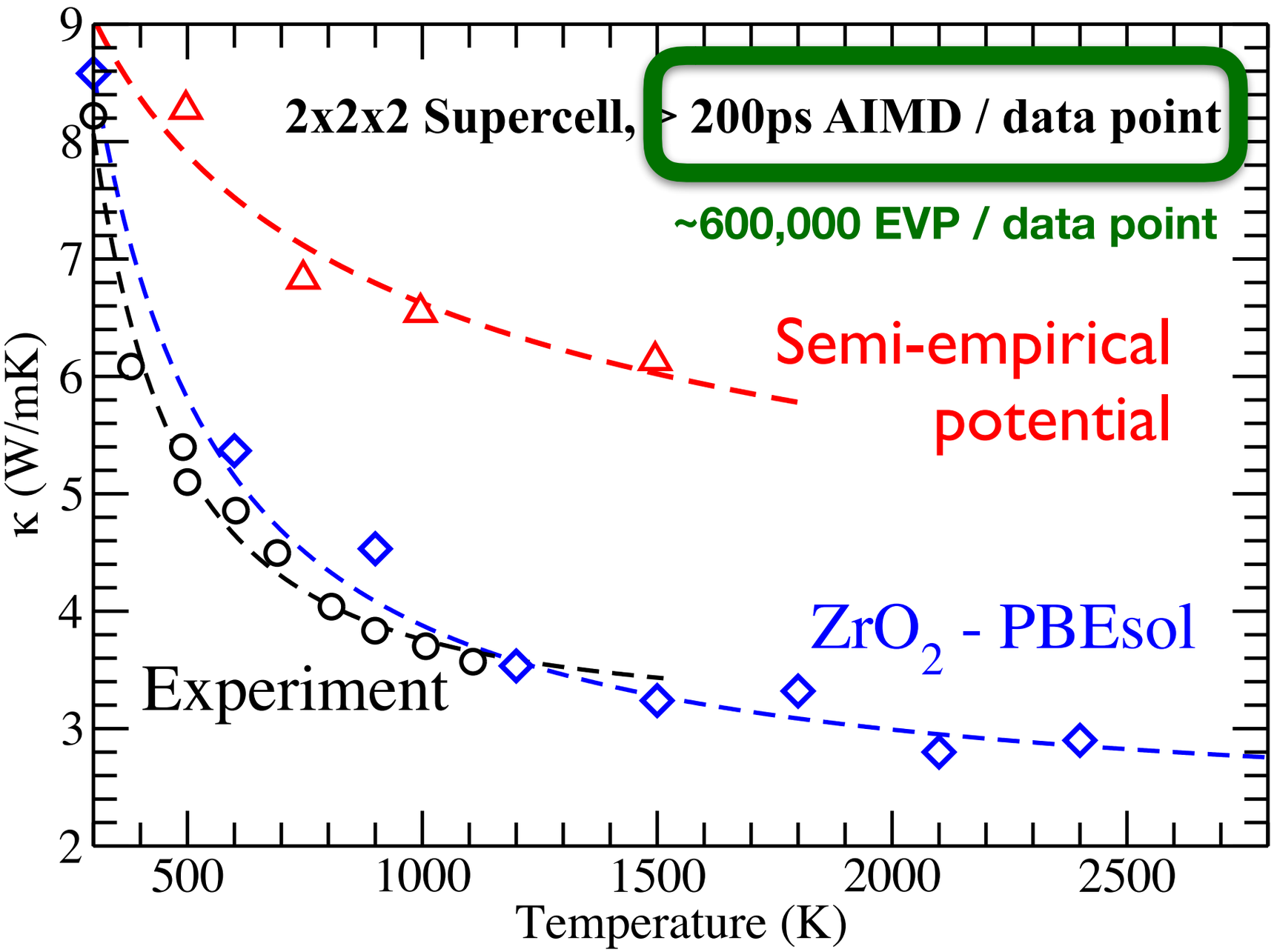
J.-F. Bisson *et al.*, *J. Am. Cer. Soc.* **83**, 1993 (2000).
G. E. Youngblood *et al.*, *J. Am. Cer. Soc.* **71**, 255 (1988).
S. Raghavan *et al.*, *Scripta Materialia* **39,** 1119 (1998).

Semi-empirical MD:

P. K. Schelling, and S. R. Phillpot,
J. Am. Cer. Soc. **84**, 2997 (2001).

THERMAL CONDUCTIVITY OF ZIRCONIA

C. Carbogno, R. Ramprasad, and M. Scheffler, *Phys. Rev. Lett.* **118**, 175901 (2017).



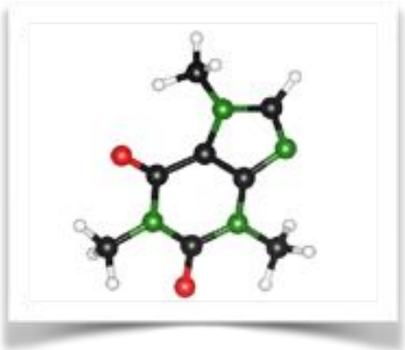
Experiment:

J.-F. Bisson *et al.*, *J. Am. Cer. Soc.* **83**, 1993 (2000).
G. E. Youngblood *et al.*, *J. Am. Cer. Soc.* **71**, 255 (1988).
S. Raghavan *et al.*, *Scripta Materialia* **39,** 1119 (1998).

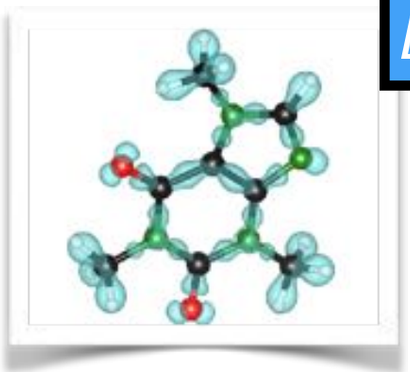
Semi-empirical MD:

P. K. Schelling, and S. R. Phillpot,
J. Am. Cer. Soc. **84**, 2997 (2001).

Update Geometry



Geometry & Species



Electronic Structure Theory

Electronic Structure

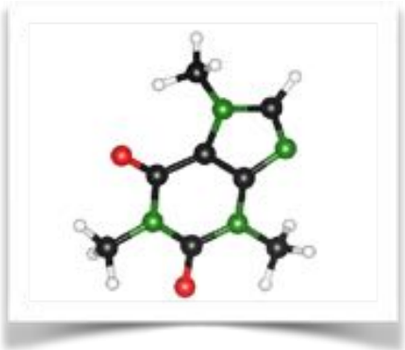
Compute Energy *E* and Forces *F* on Atoms

⇒ Materials' Properties

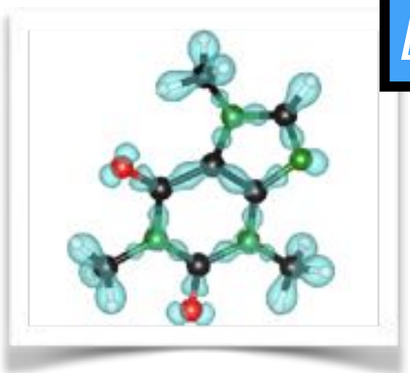
A series of **different**, but **similar** **eigenvalue problems** are solved one after another.

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

Update Geometry



Geometry & Species



Electronic Structure Theory

Electronic Structure

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

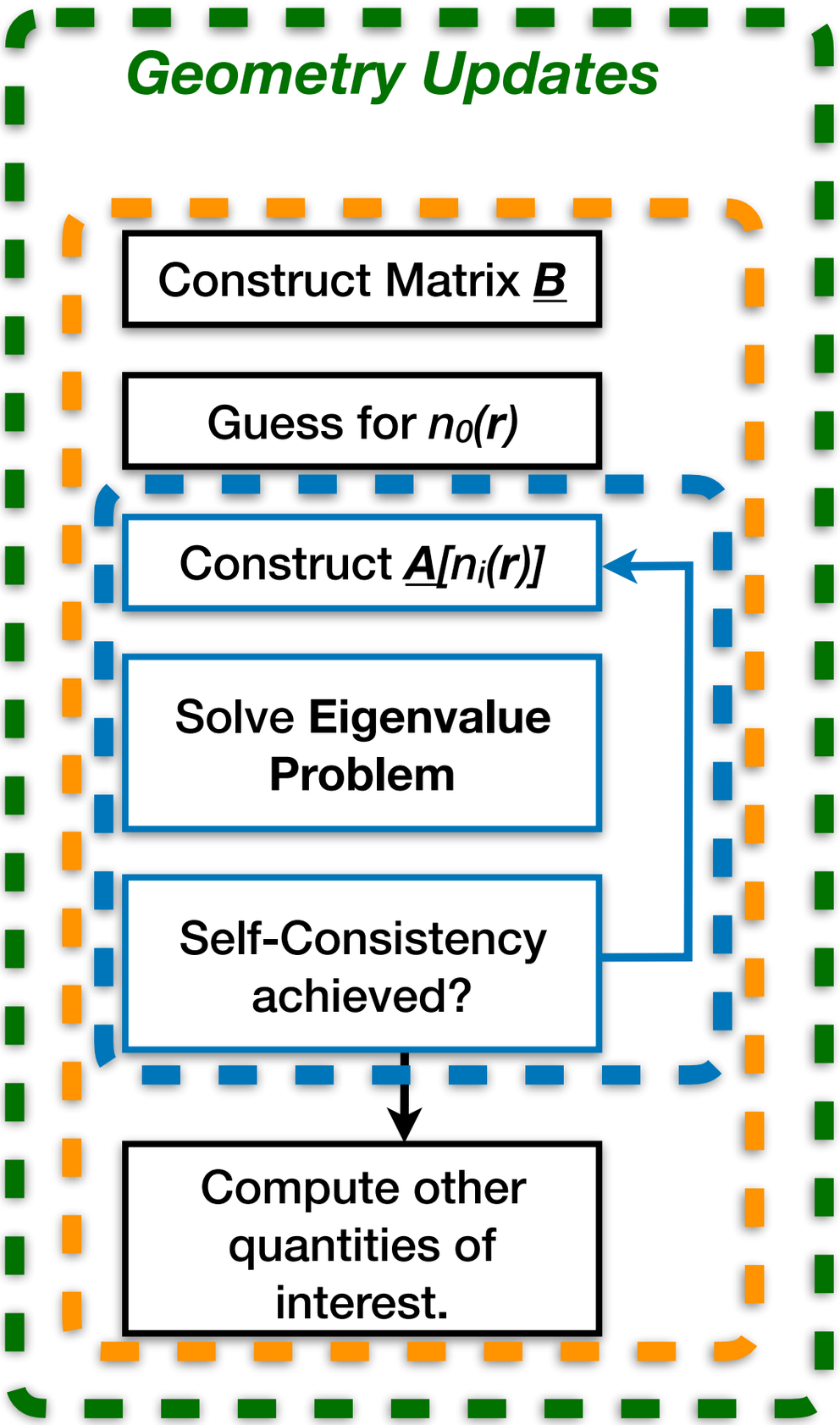
⇒ Materials' Properties

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:



These are **several** self-concistency 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

```
! Solve evp, return eigenvalues and eigenvectors
if(ph%elpa_solver == 2) then
   success = elpa_solve_evp_real_2stage_double(ph%n_good,ph%n_states_solve,&
                ham,bh%n_lrow,eval,evec,bh%n_lrow,bh%blk,bh%n_lcol,&
                ph%elpa_comm_row,ph%elpa_comm_col,bh%comm)
else
   success = elpa_solve_evp_real_1stage_double(ph%n_good,ph%n_states_solve,&
                ham,bh%n_lrow,eval,evec,bh%n_lrow,bh%blk,bh%n_lcol,&
                ph%elpa_comm_row,ph%elpa_comm_col,bh%comm)
endif
```

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**

```
! Create ELPA object
elpa => elpa_allocate()

! Set MANDATORY parameters decribing the matrix and it's MPI distribution
call elpa%set("na", na, success)
call elpa%set("nev", nev, success)
call elpa%set("local_nrows", na_rows, success)
call elpa%set("local_ncols", na_cols, success)
call elpa%set("nblk", nblk, success)
call elpa%set("mpi_comm_parent", mpi_comm_world, success)
call elpa%set("process_row", my_prow, success)
call elpa%set("process_col", my_pcol, success)
success = elpa%setup()

! Set OPTIONAL run-time options
call elpa%set("solver", elpa_solver_2stage, success)

! Use ELPA
call elpa%eigenvectors(a, ev, z, success)
~
```

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**

```
! Create ELPA object
elpa => elpa_allocate()

! Set MANDATORY parameters decribing the matrix and it's MPI distribution
call elpa%set("na", na, success)
call elpa%set("nev", nev, success)
call elpa%set("local_nrows", na_rows, success)
call elpa%set("local_ncols", na_cols, success)
call elpa%set("nblk", nblk, success)
call elpa%set("mpi_comm_parent", mpi_comm_world, success)
call elpa%set("process_row", my_prow, success)
call elpa%set("process_col", my_pcol, success)
success = elpa%setup()

! Set OPTIONAL run-time options
call elpa%set("solver", elpa_solver_2stage, success)

! Use ELPA
call elpa%eigenvectors(a, ev, z, success)
~
```

ELPA is able to store and exploit information across calls.

Example: Autotuning

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

```
! Setup ELPA using the new API
succes = 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 cal
  call elpa%eigenvectors(a, ev, z, error)

enddo
```

Example: Autotuning

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

```
! Setup ELPA using the new API
succes = 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 cal
  call elpa%eigenvectors(a, ev, z, error)

enddo
```

Manual Settings or Predefined Options

Example: Autotuning

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

```
! Setup ELPA using the new API
succes = 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 cal
  call elpa%eigenvectors(a, ev, z, error)

enddo
```

Manual Settings or Predefined Options

Autotuning can be used during production iterations.

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

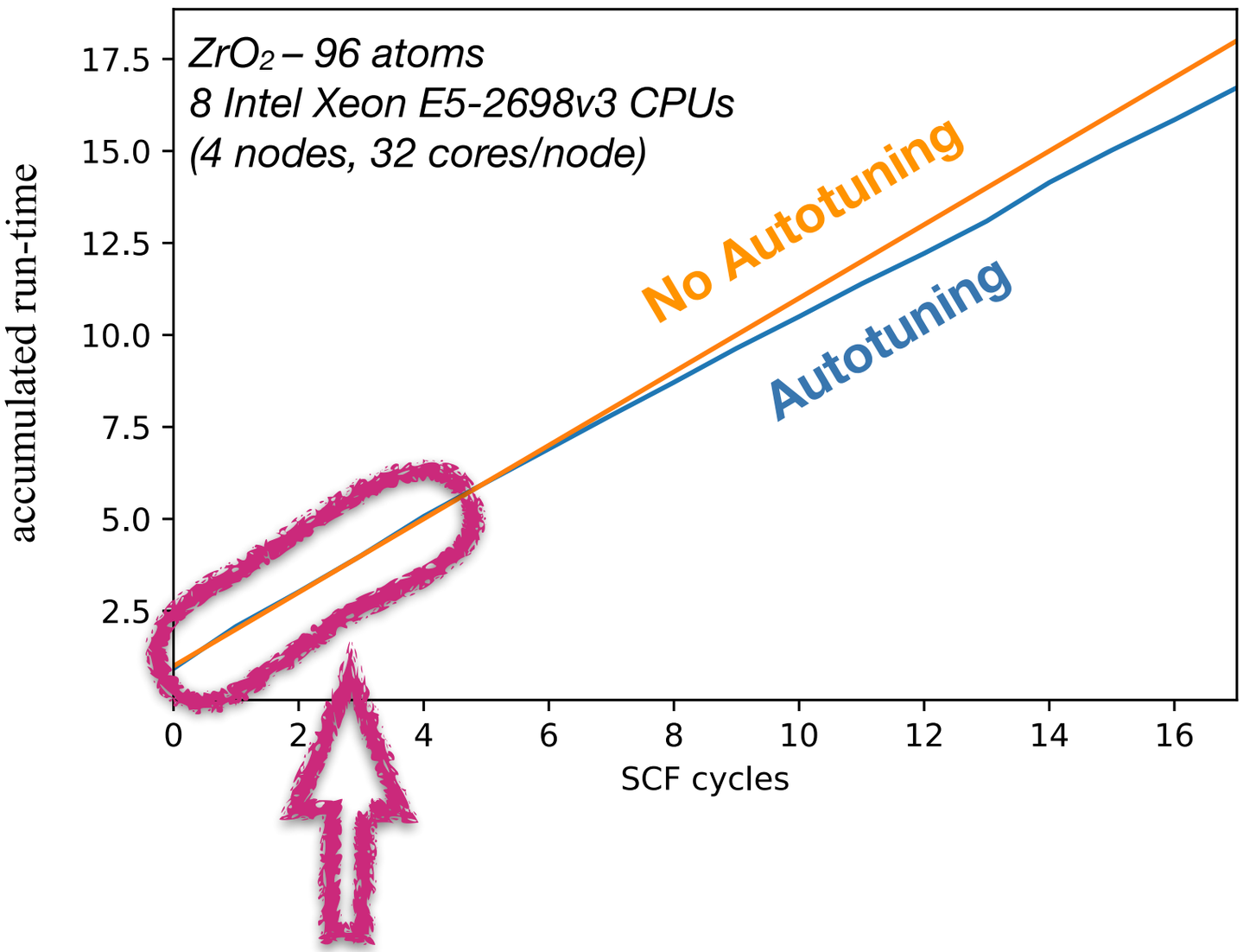
GPU offload:
Which routines
should make use
of GPUs?

ELPA_AUTOTUNE_FAST

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

Autotuning with ELPA_AUTOTUNE_FAST

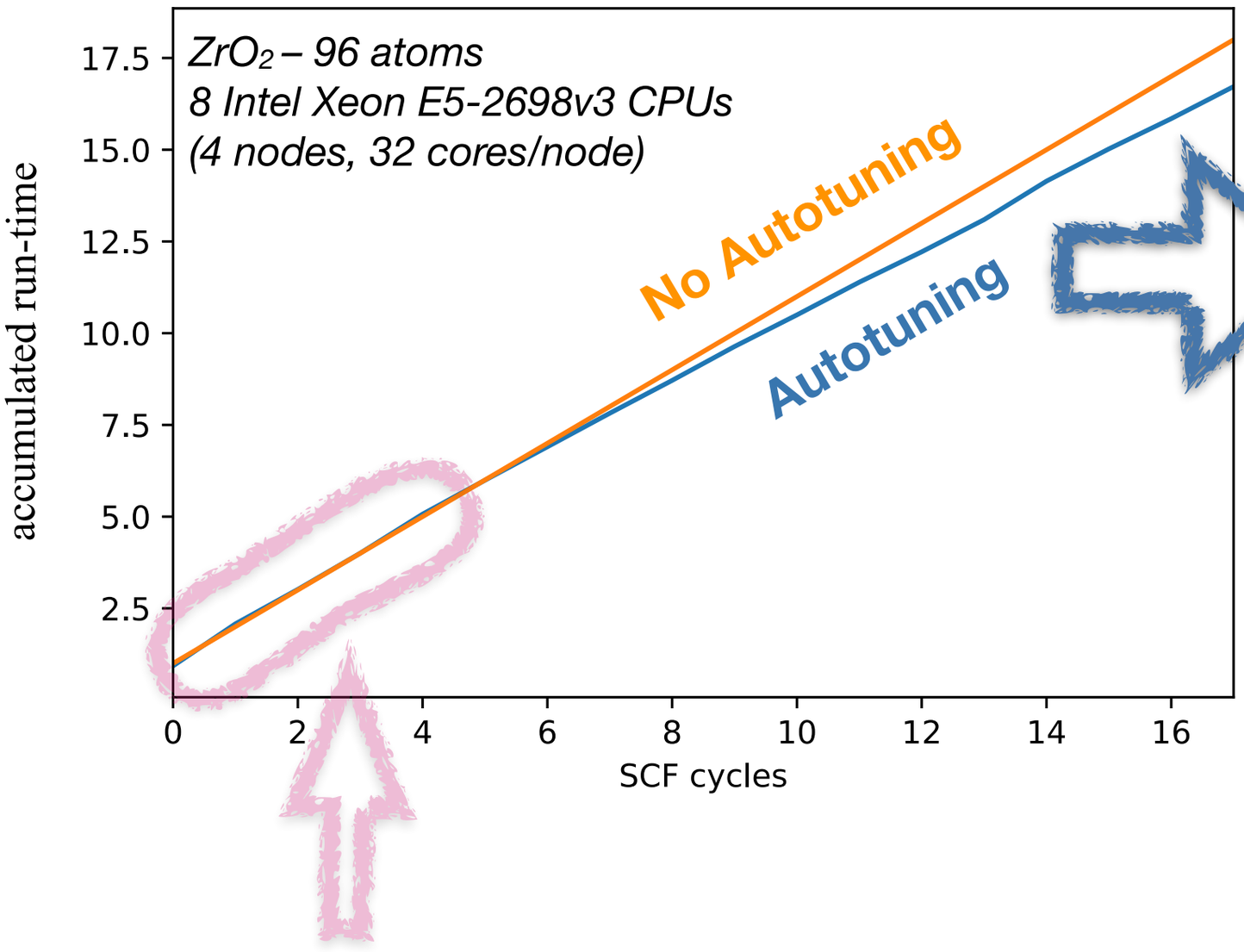
Accumulated Runtimes:



Slight computational overhead observed in the **first couple** of iterations…

Autotuning with ELPA_AUTOTUNE_FAST

Accumulated Runtimes:

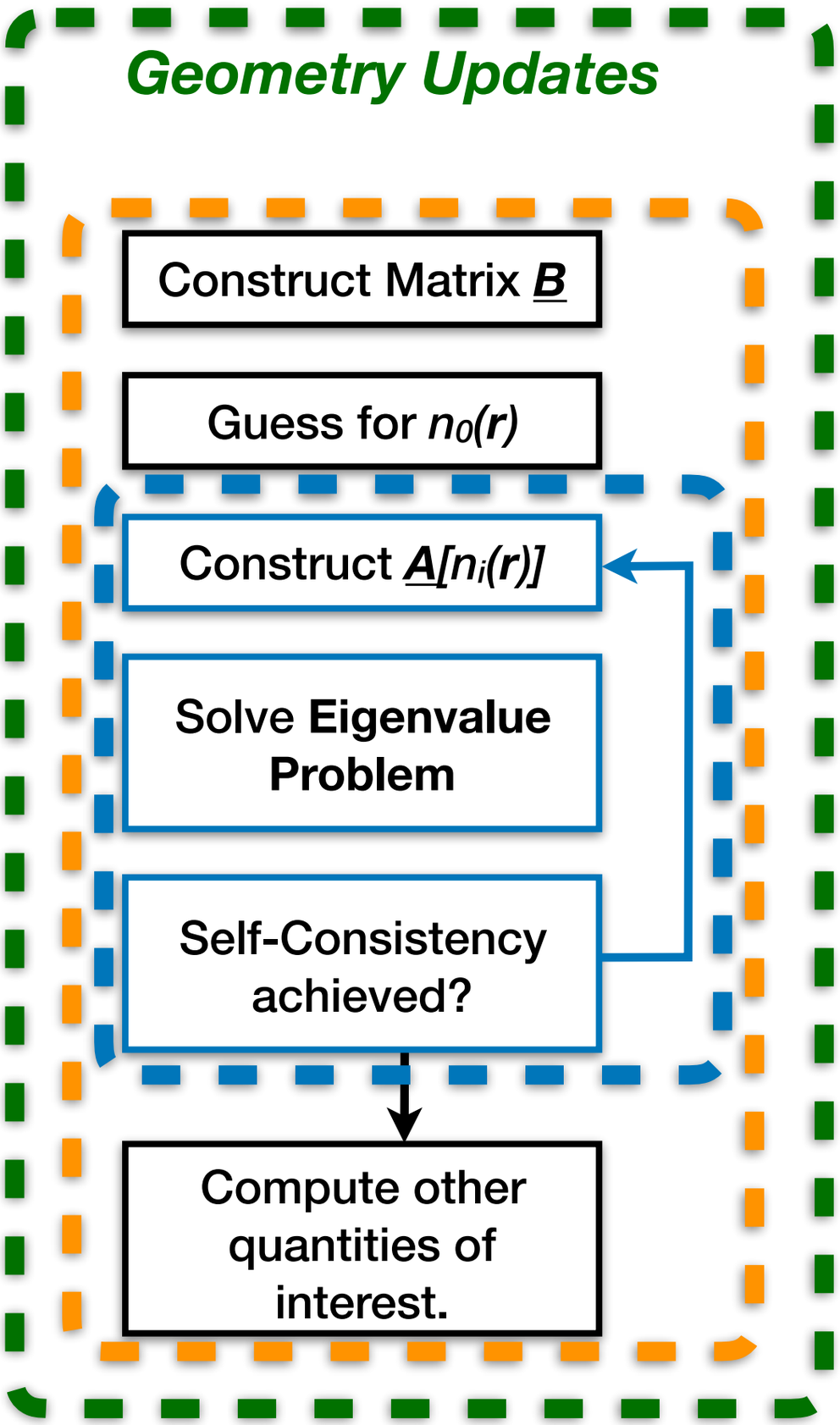


Slight computational overhead observed in the first couple of iterations…

Autotuning beneficial in the long-run

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

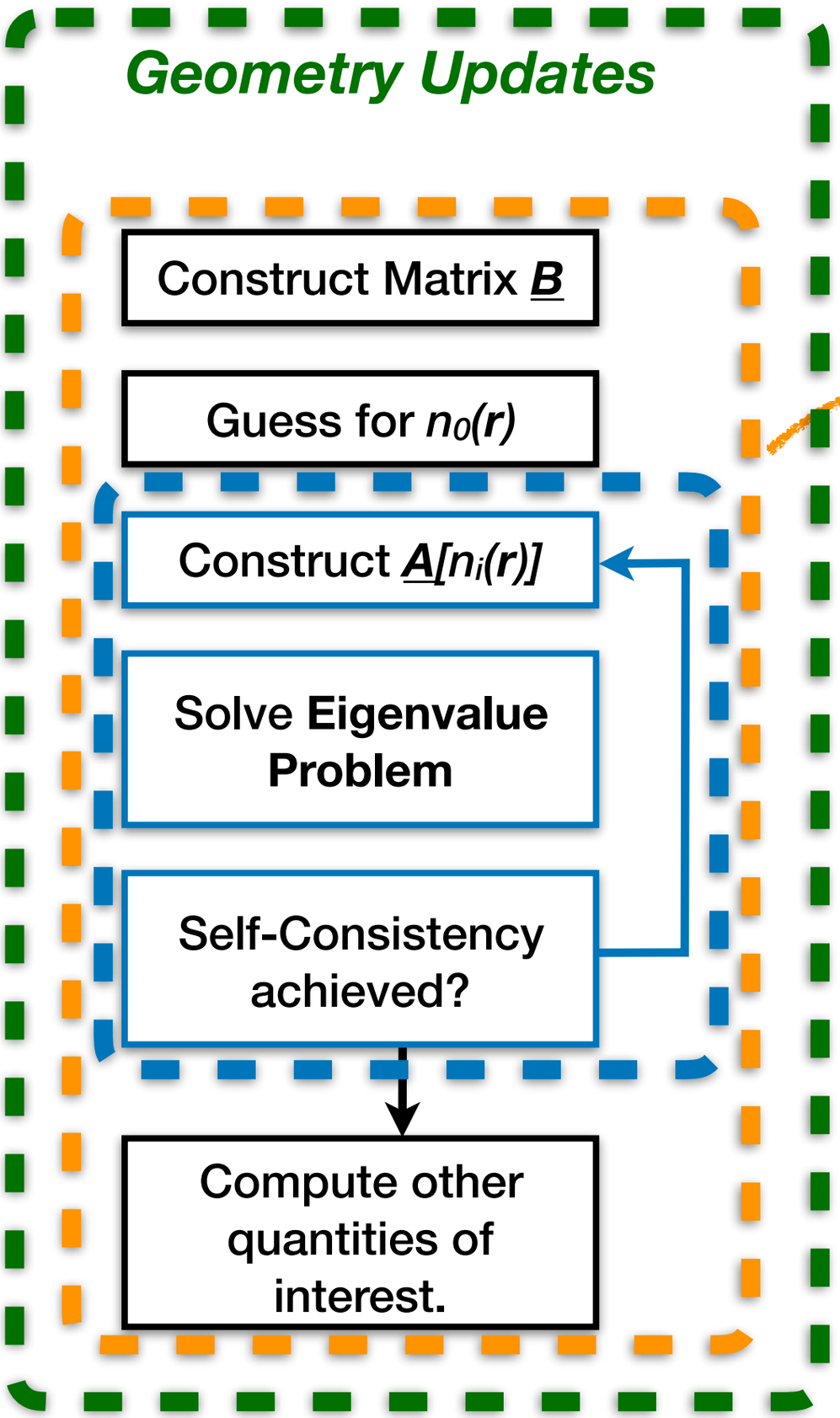
Overview:



These are **several** self-concistency cycles.

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

Overview:

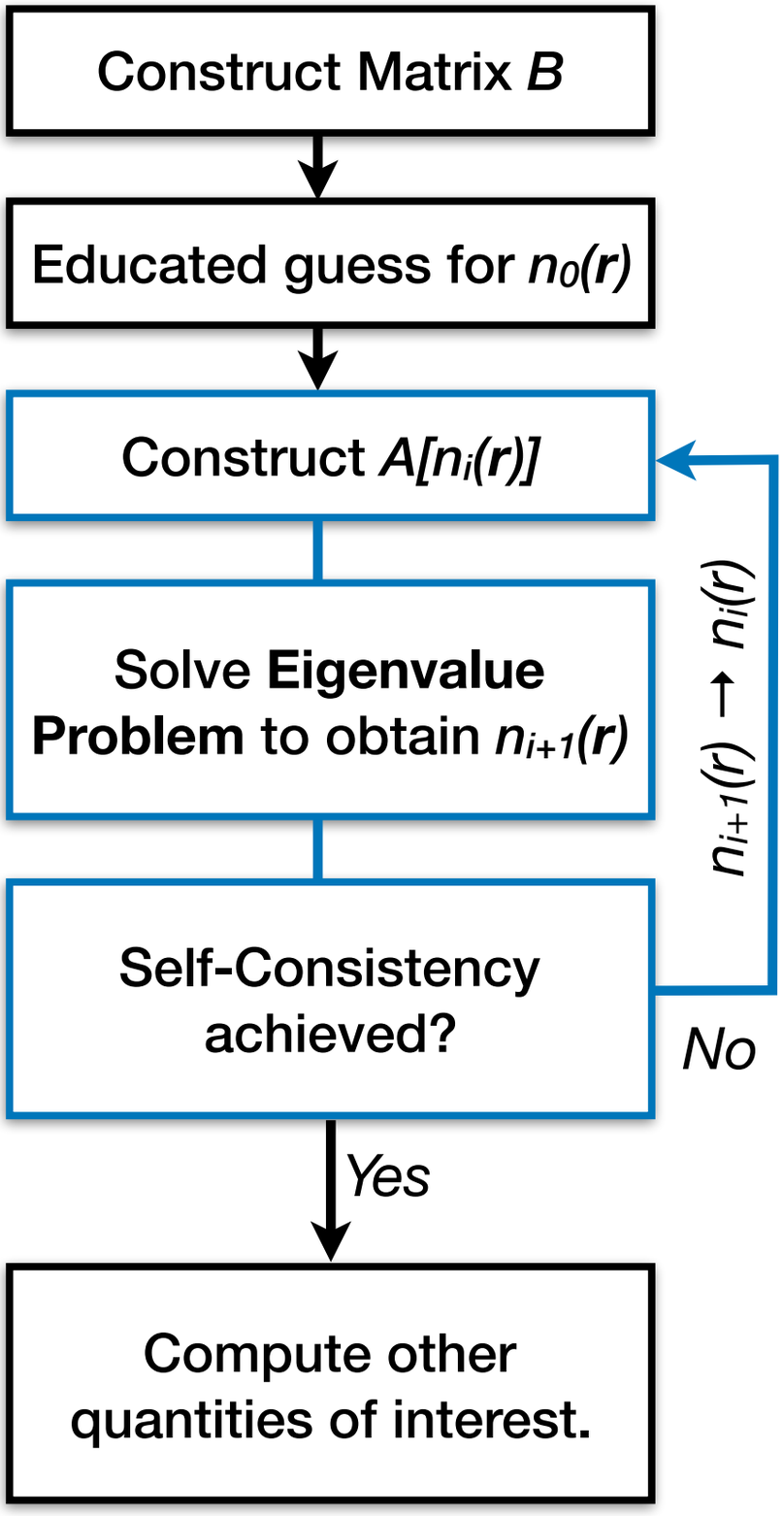


These are **several** self-concistency cycles.

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

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

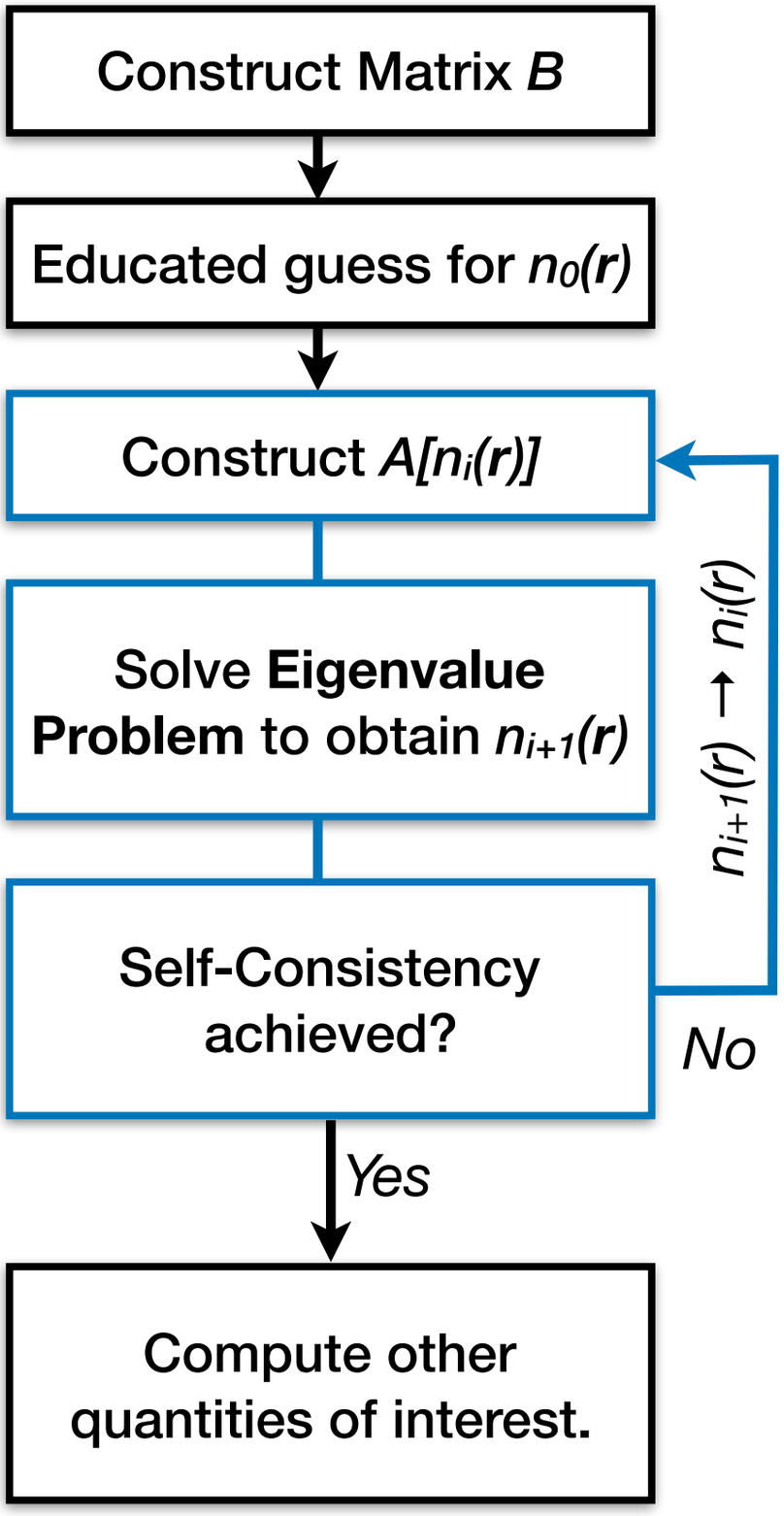
The Self-Consistency Cycle



Some Remarks:

- **Typical number of iterations:** 10-100
- **Convergence accelerated by mixing schemes**
- **Convergence/Self-concistency monitored via:**
  - Change in **density** $n_i(\mathbf{r})$
  - Change in **energy** $E_i(\mathbf{r})$
  - Change in **forces** $\boldsymbol{F_I}$
  - Change in **eigenenergies** ε_i

The Self-Consistency Cycle



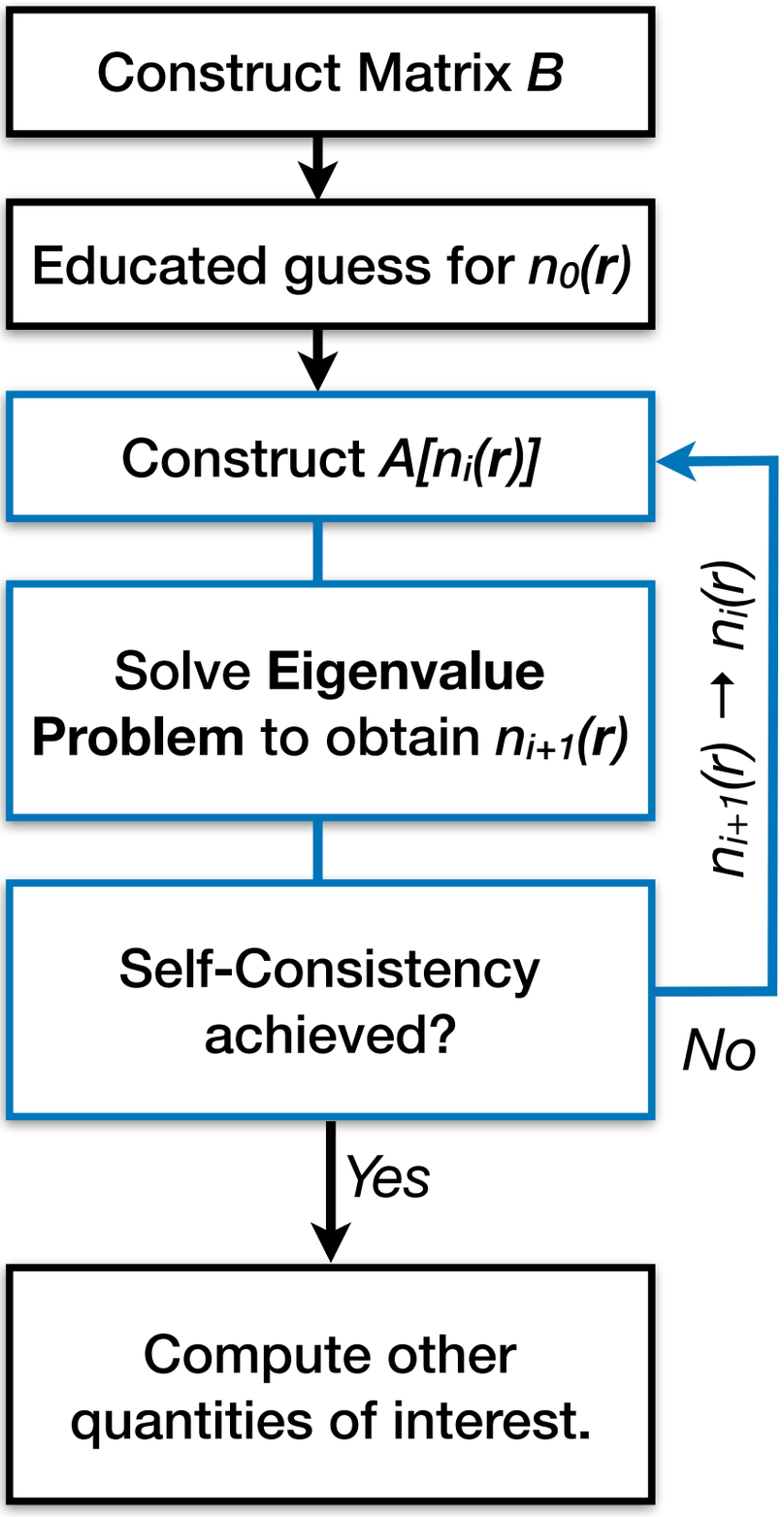
Some Remarks:

- **Typical number of iterations:** 10-100
- **Convergence accelerated by mixing schemes**
- **Convergence/Self-concistency monitored via:**
  - Change in **density** $n_i(r)$
  - Change in **energy** $E_i(r)$

  Extensive & Variational
  - Change in **forces** F_I
  - Change in **eigenenergies** ε_i

  Intensive & Non-Variational

The Self-Consistency Cycle



Some Remarks:

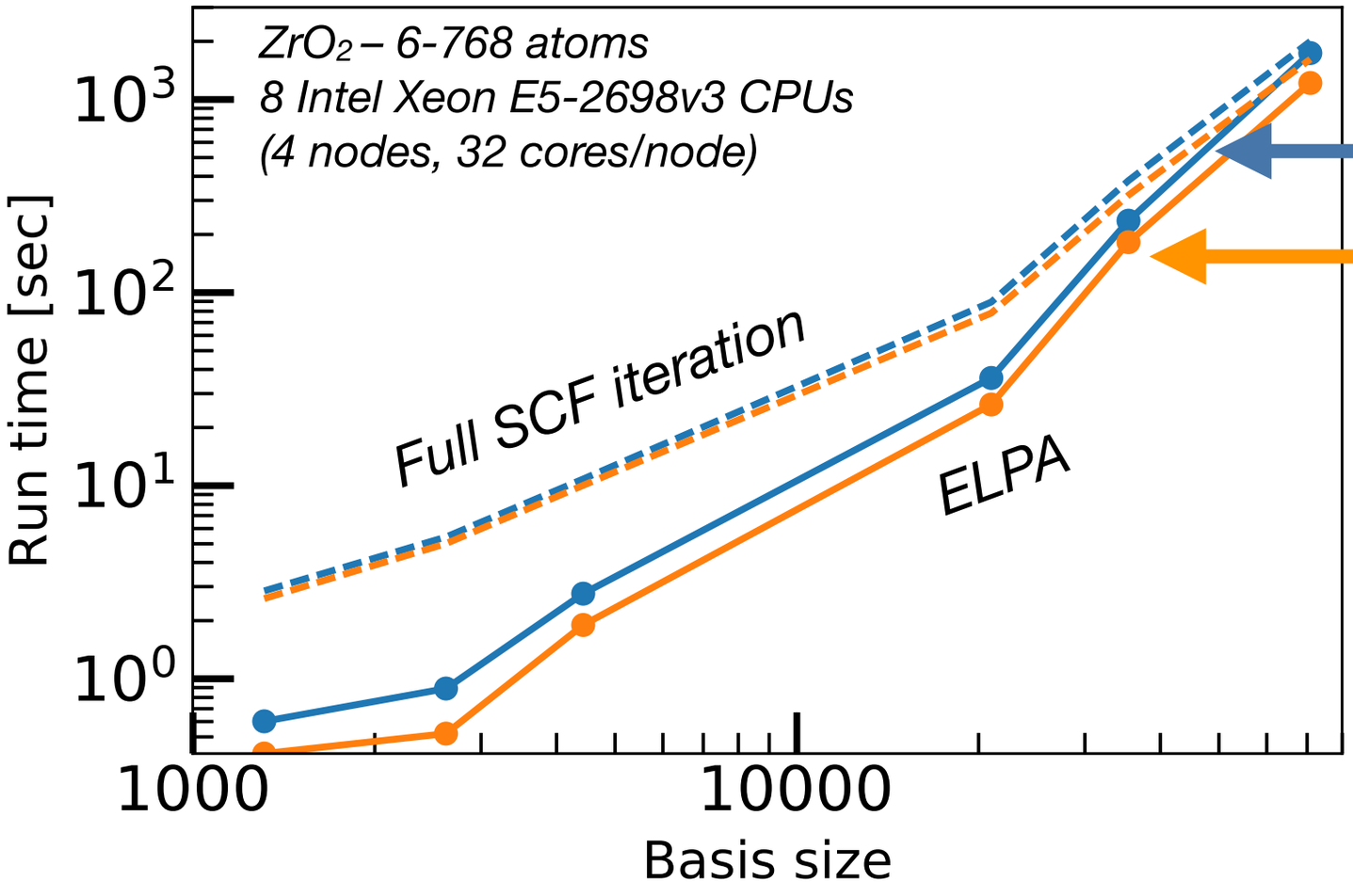
- **Typical number of iterations:** 10-100
- **Convergence accelerated by mixing schemes**
- **Convergence/Self-concistency monitored via:**
  - Change in **density** $n_i(r)$
  - Change in **energy** $E_i(r)$

  Extensive & Variational
  - Change in **forces** F_I
  - Change in **eigenenergies** ε_i

  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

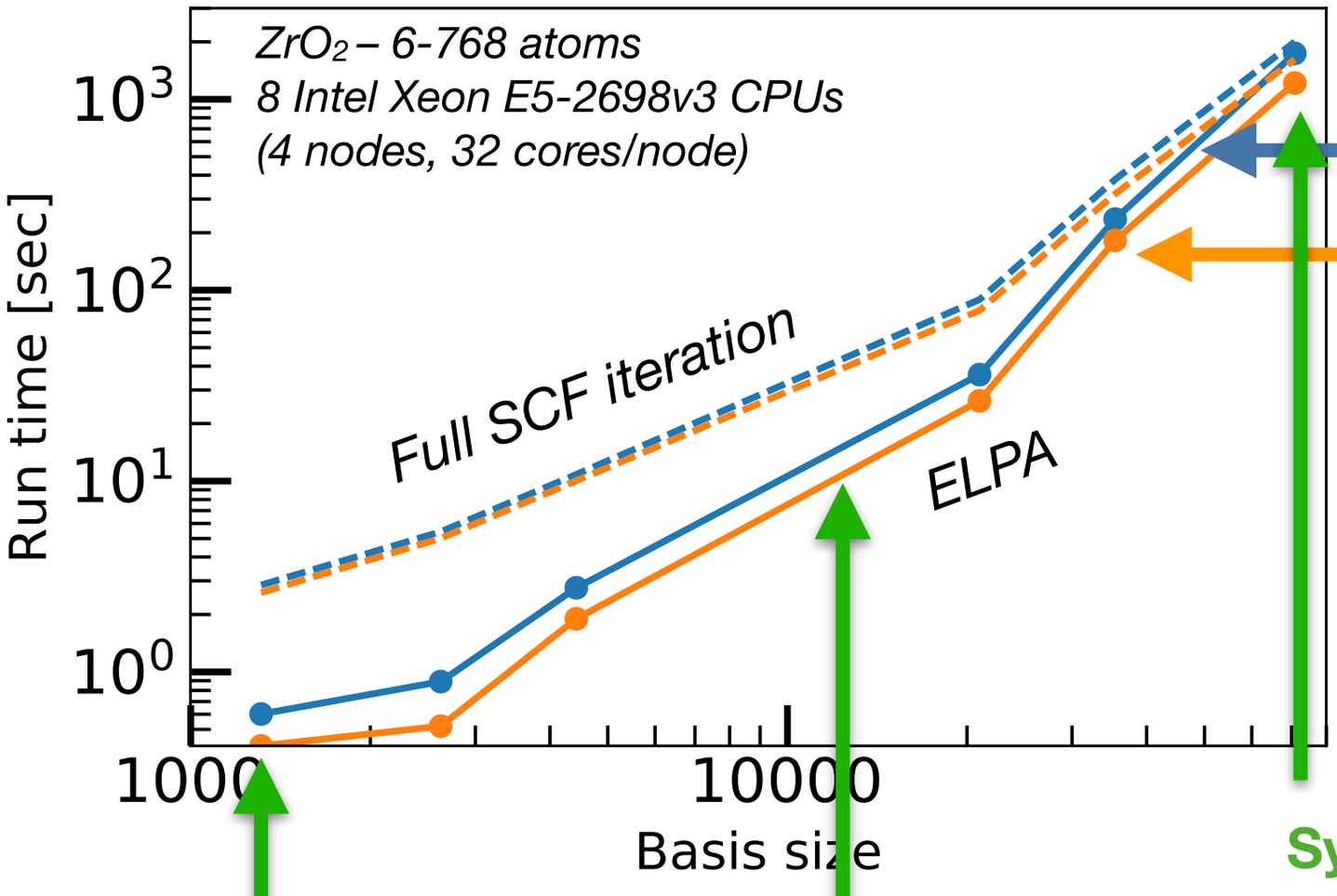


Standard double precision (DP)

Single Precision (SP)

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

Mixed Precision Calculations



Standard double precision (DP)

Single Precision (SP)

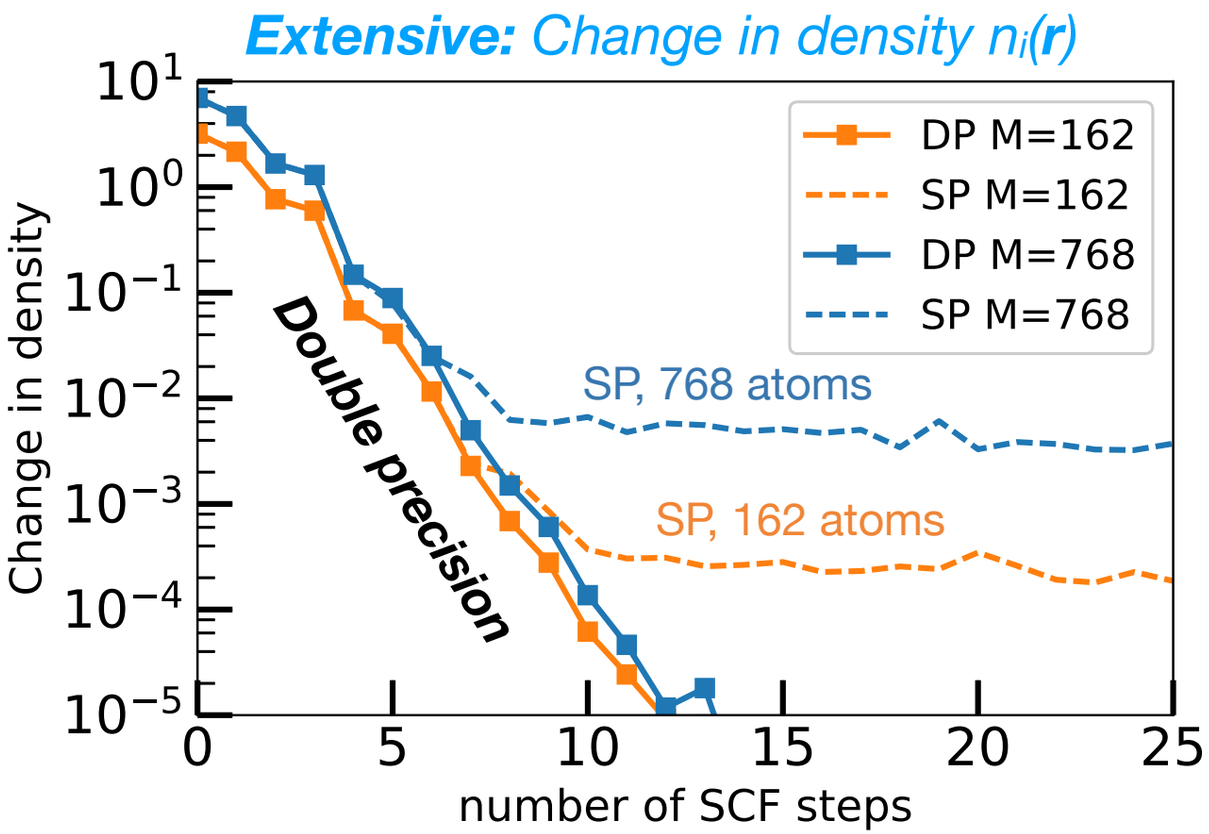
- **SP** routines **~20%** faster
- **Eigenvalue problem** dominates comp. cost for larger systems.

System size needed to model **atomic defects** in the **crystal.**

System size needed for a **thermal conductivity** calculation in **pristine** ZrO_2.

System size needed for a **relaxation** (“PES minima search”).

How far can we go?

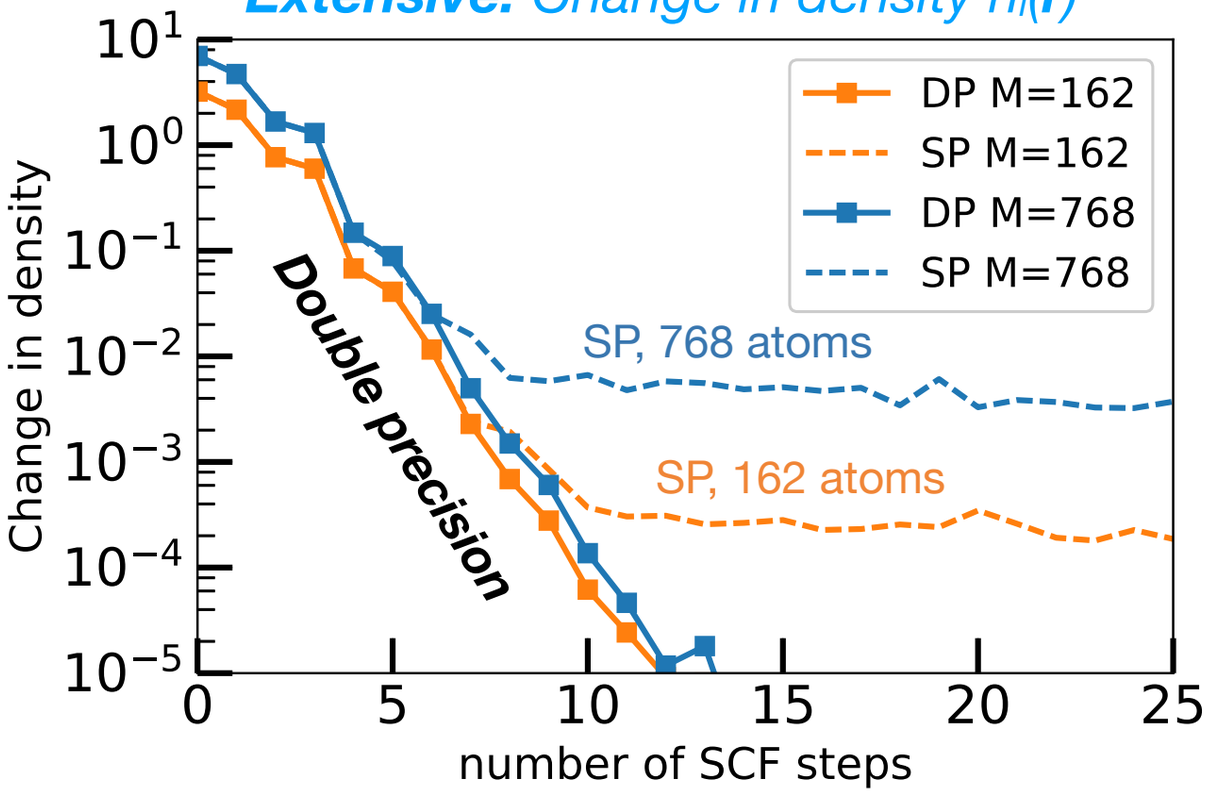


Monitoring the convergence of the self-consistency cycle

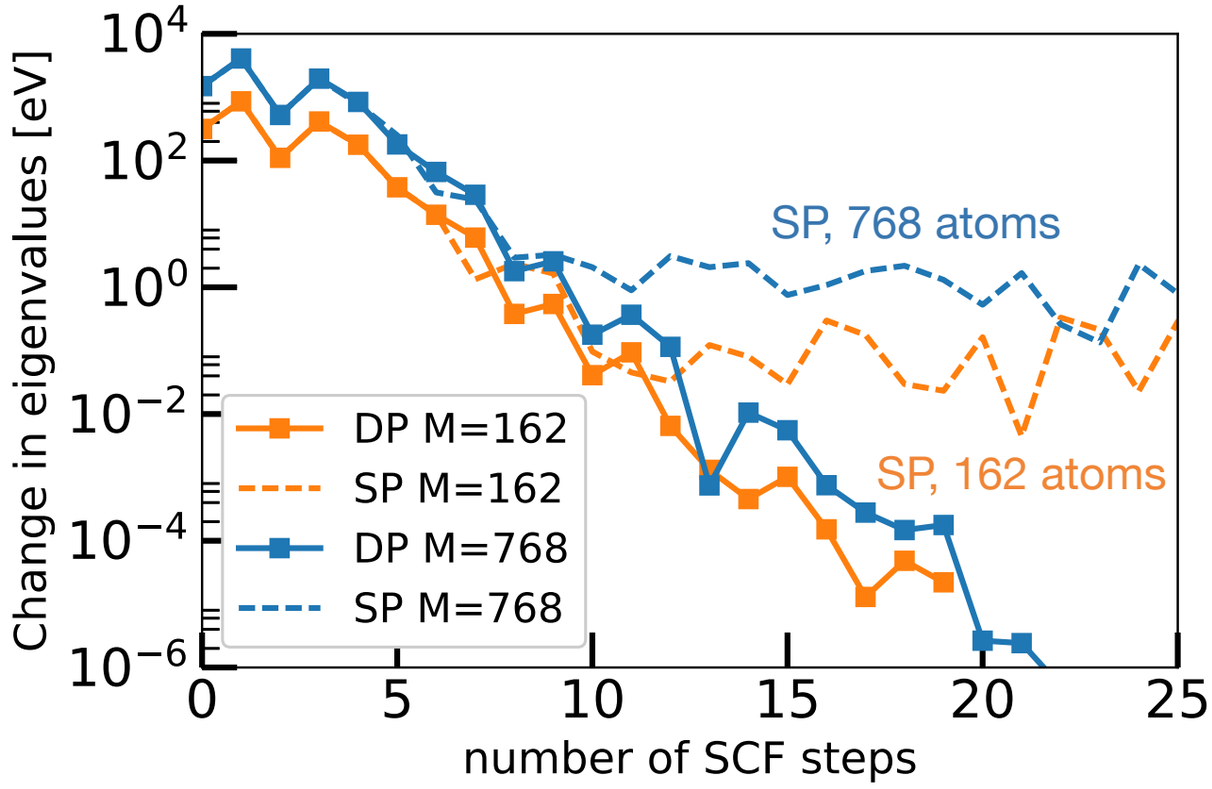
- **Single-Precision** routines **do not allow** to reach **convergence.**
- ***"First"*** few iterations **can be performed** with **single-precision** without loss of accuracy.
- Break-down point of **single-precision** routines system-size-dependent.

How far can we go?

Extensive: *Change in density $n_i(\mathbf{r})$*



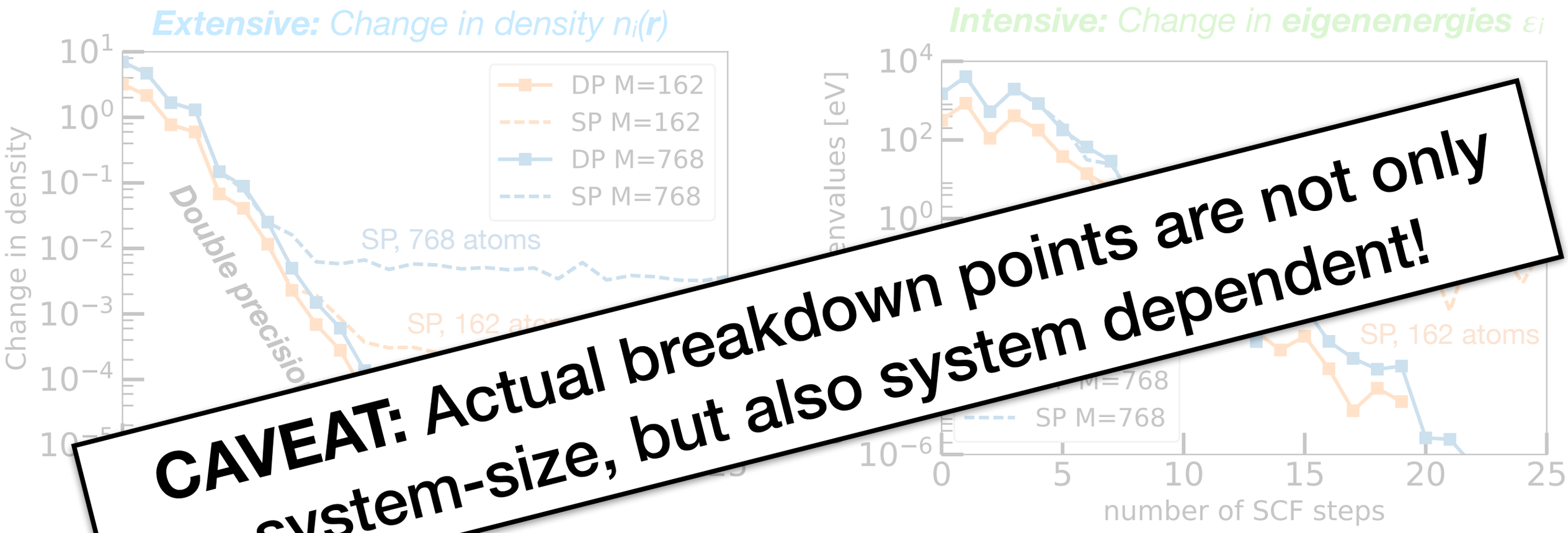
Intensive: *Change in* ***eigenenergies*** ε_i



- **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?

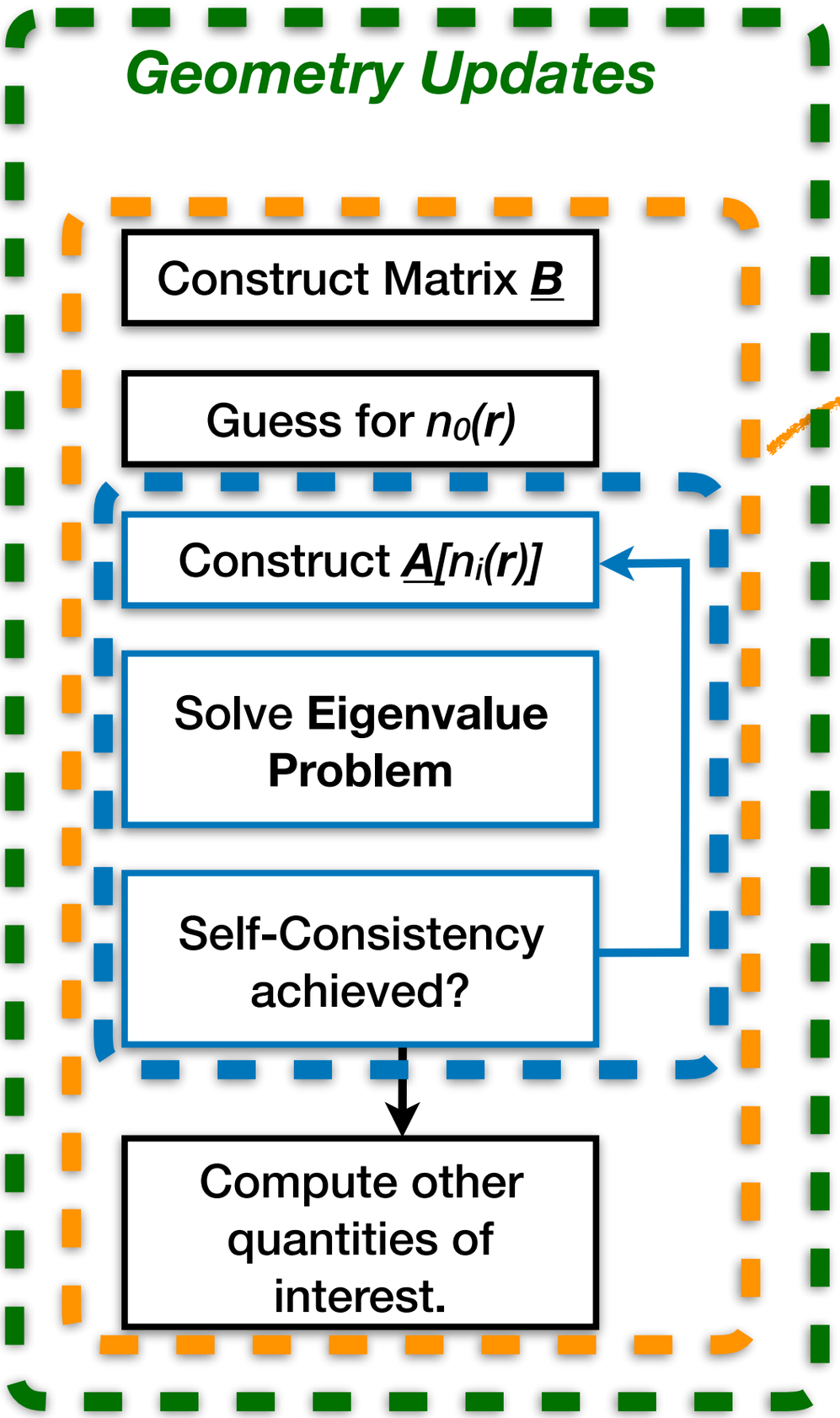


CAVEAT: Actual breakdown points are not only system-size, but also system dependent!

- **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.**

Overview:



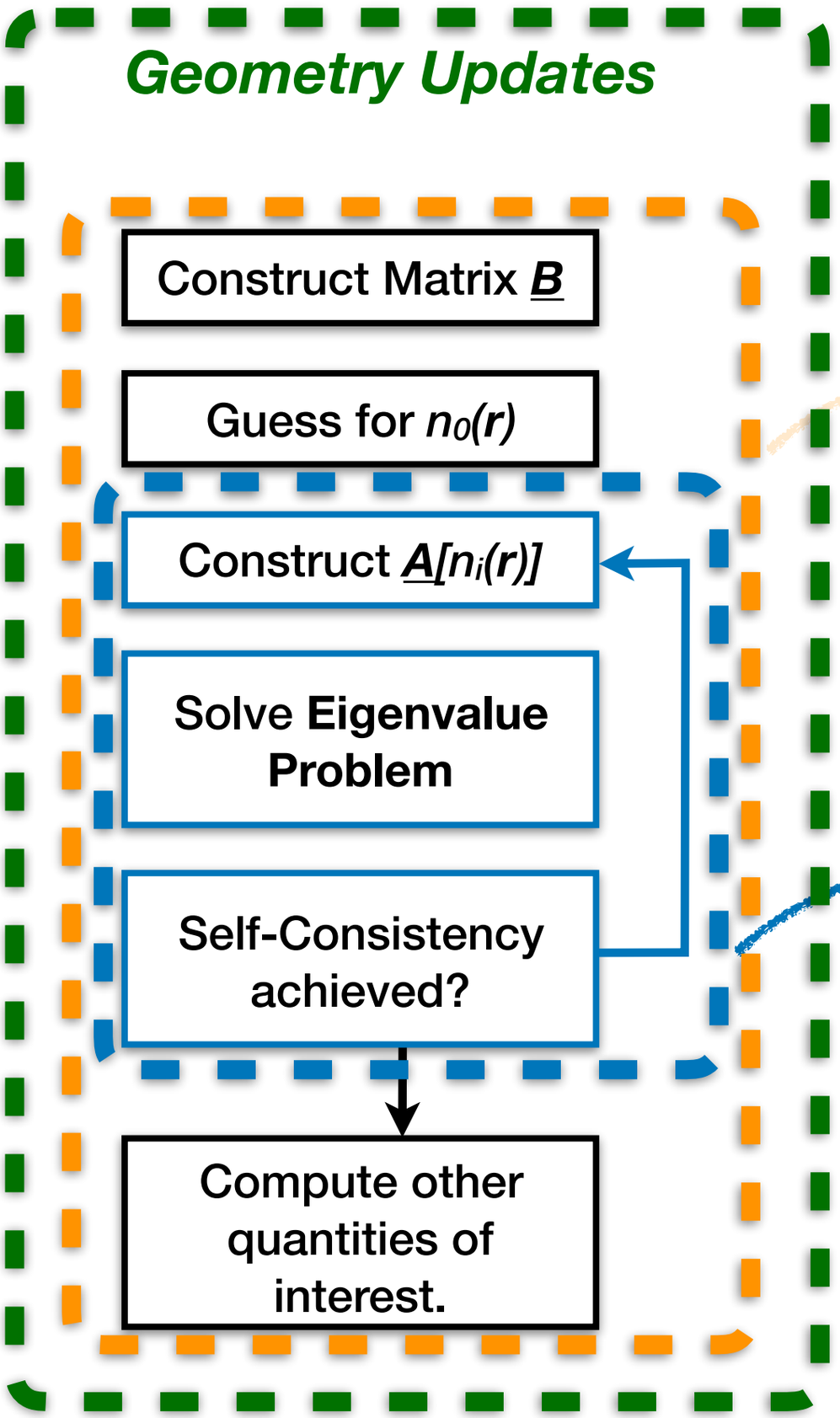
These are **several** self-concistency cycles.

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

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

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

Overview:



These are **several** self-concistency cycles.

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

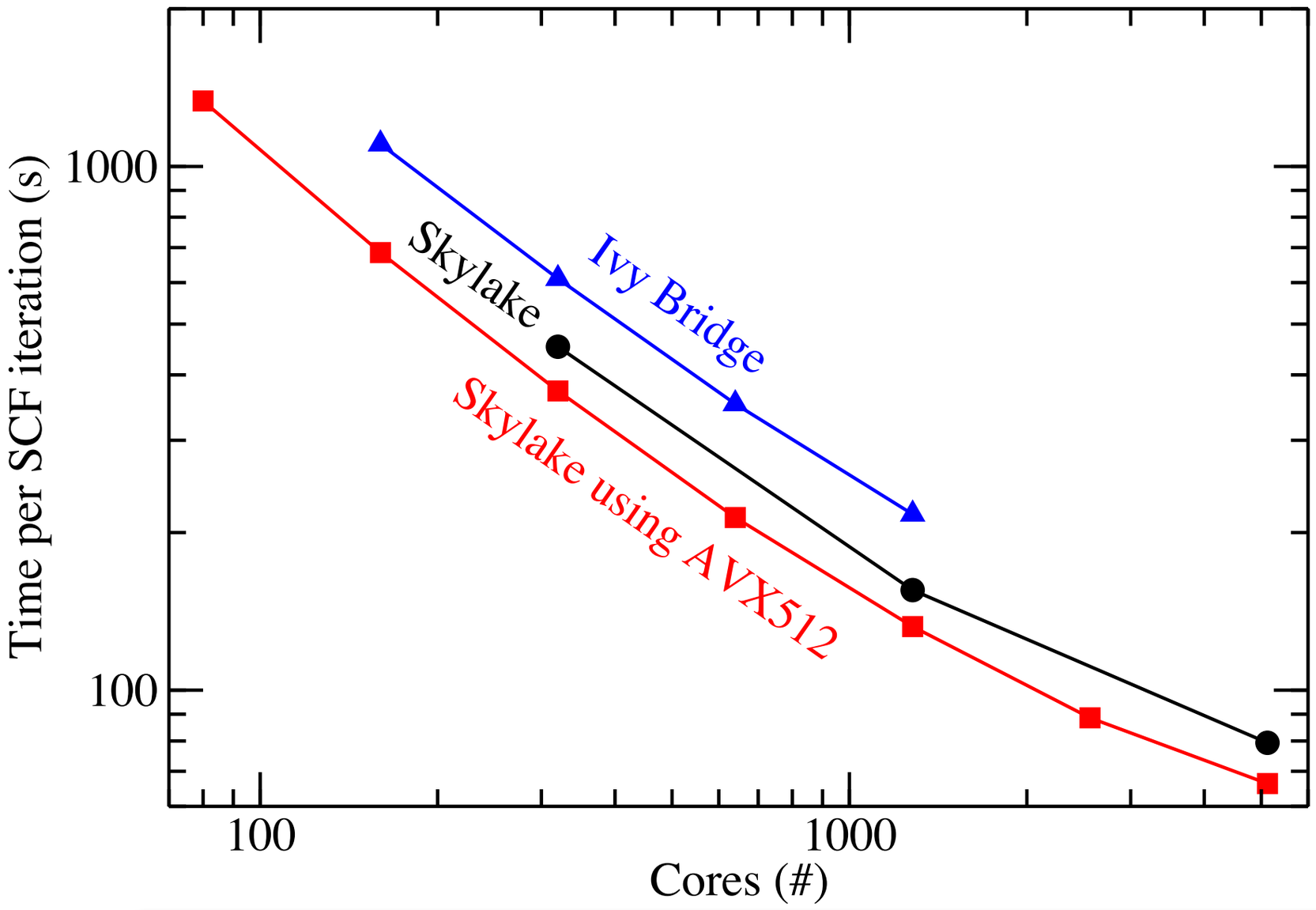
This is **one** self-concistency cycle.
(**Several** Eigenvalue Problems)

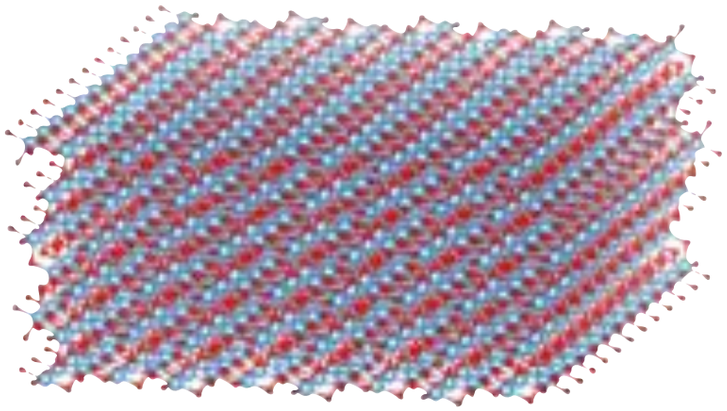
⇒ **(B)** Exploit **Single/Double Precision routines**
in ELPA-AEO for **a series** of Eigenvalue Problem

This is **one** self-concistency interation.
(**One** Eigenvalue Problem)

Accelerating One Iteration

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



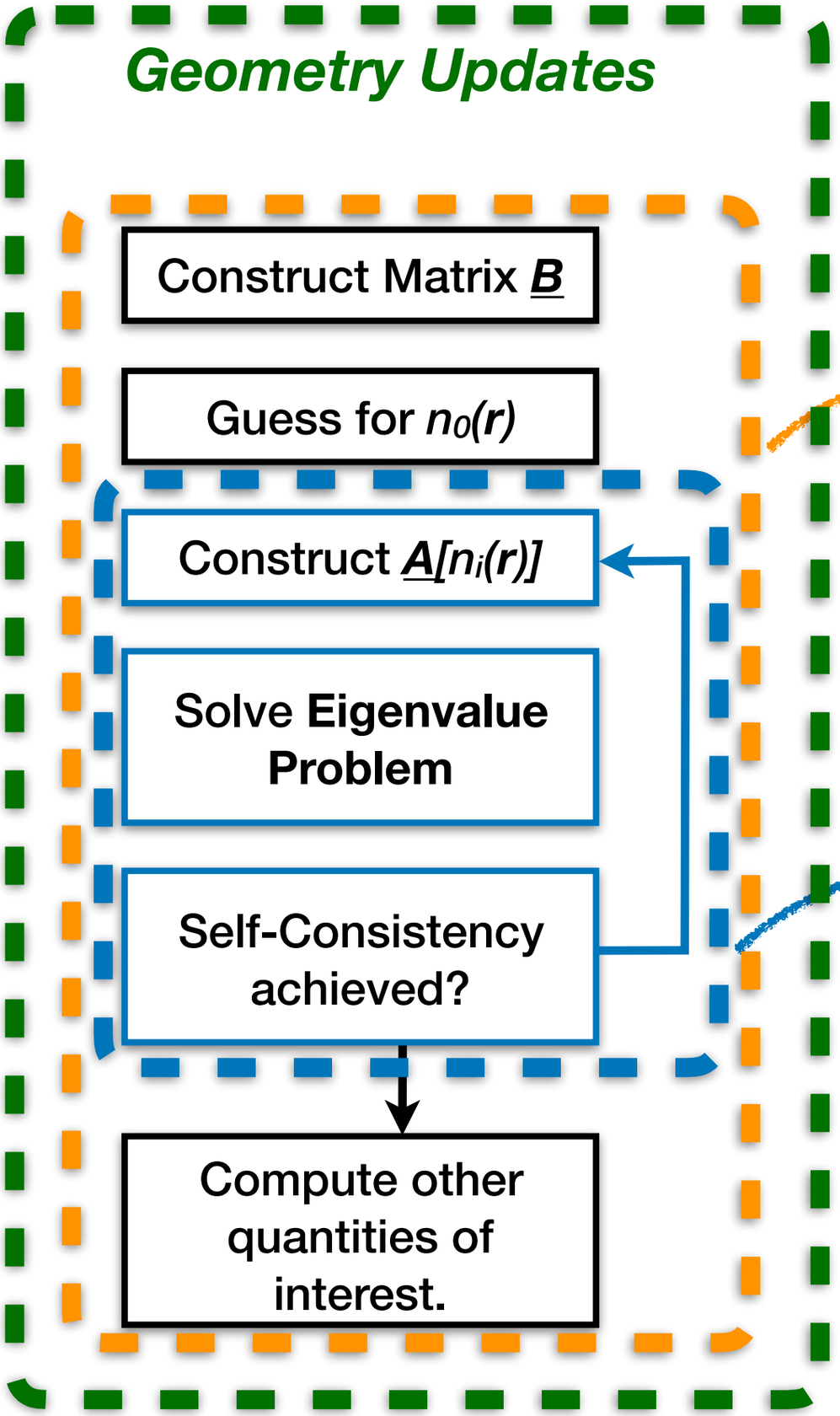


TiO_2 slab/surface with
3,500 atoms and
100 Å vacuum

⇒ Optimizations in ELPA

directly translate into an application speed-up.

SUMMARY:



These are **several** self-concistency cycles.

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

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

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

This is **one** self-concistency 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

BERGISCHE UNIVERSITÄT WUPPERTAL

MPCDF Max Planck Computing & Data Facility

Bundesministerium für Bildung und Forschung

Acknowledgments

Danilo Simoes Brambila
FHI

(now at some fancy Startup)

Matthias Scheffler
FHI

The ELPA-AEO Consortium

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

BERGISCHE UNIVERSITÄT WUPPERTAL

MPCDF Max Planck Computing & Data Facility

Bundesministerium für Bildung und Forschung

ELSI

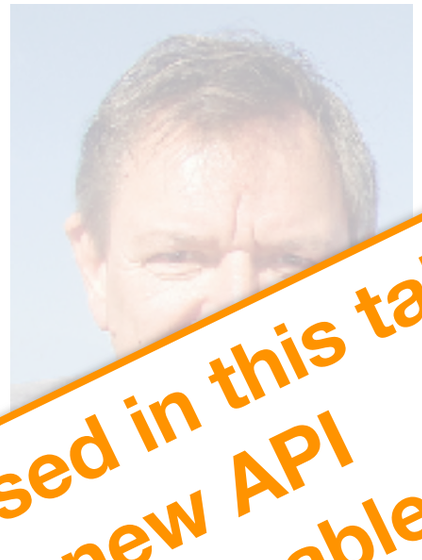
Volker Blum

Victor Yu

Acknowledgments

Danilo Simoes Brambila
FHI

(now at some fancy Startup)



Matthias Scheffler

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

The ELSI project

http://

BERGISCHE UNIVERSITÄT WUPPERTAL

MAX PLANCK COMPUTING & DATA FACILITY

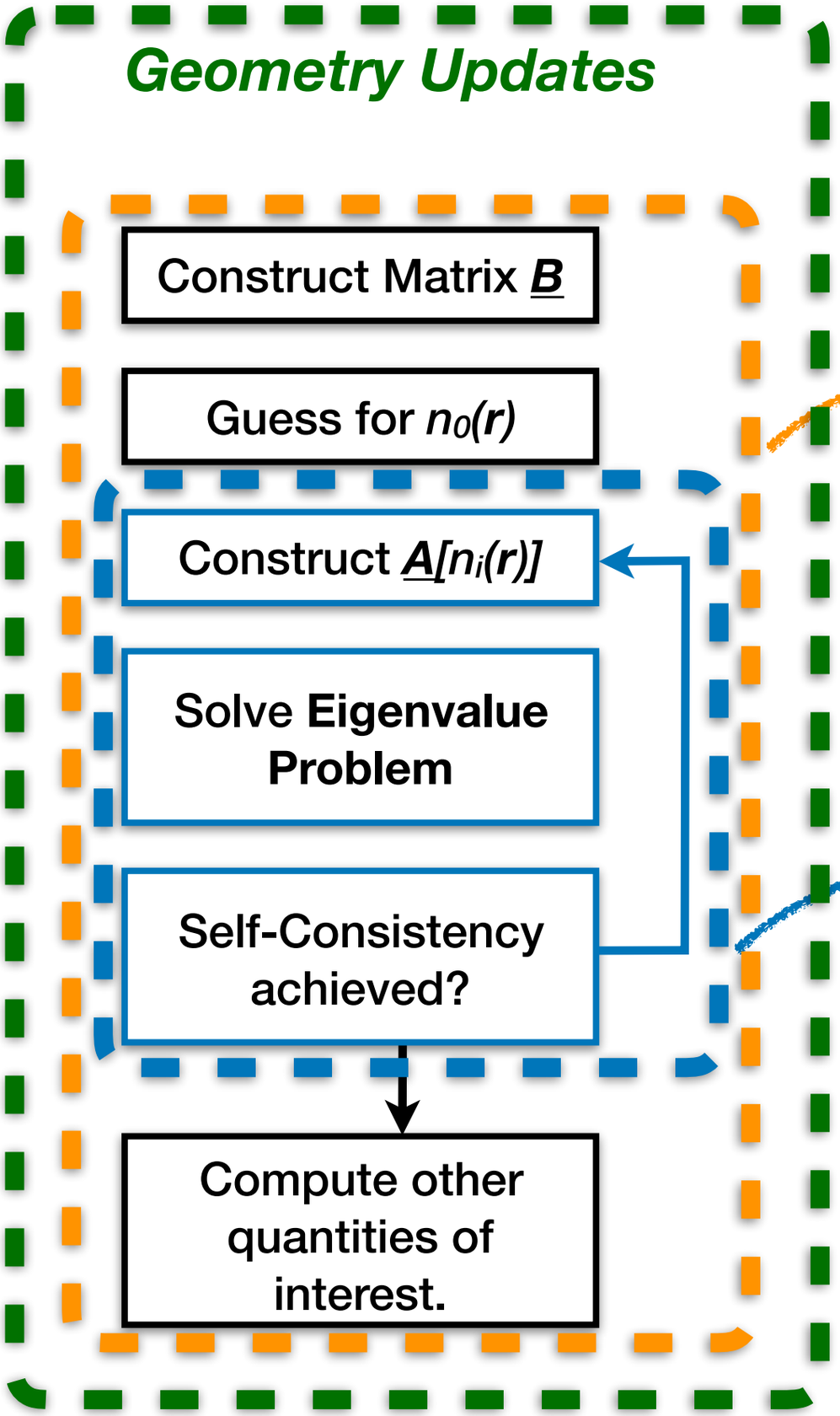
Bundesministerium für Bildung und Forschung

ELSI

Volker Blum

Victor Yu

SUMMARY:



These are **several** self-concistency cycles.

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

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

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

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

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