A Tutorial

for the FHI-aims UK meeting

Sebastian Kokott
Konstantin Lion
Andrey Sobolev
James Green
Uthpala Herath
Min-ye Zhang
Volker Blum

THE MS1P TEAM – Warwick – May 15, 2024

Overview

How this tutorial is structured

Two main sessions:

A. Hybrid density functionals
B. Beyond DFT: GW method for molecules and solids

Each Session has two parts:

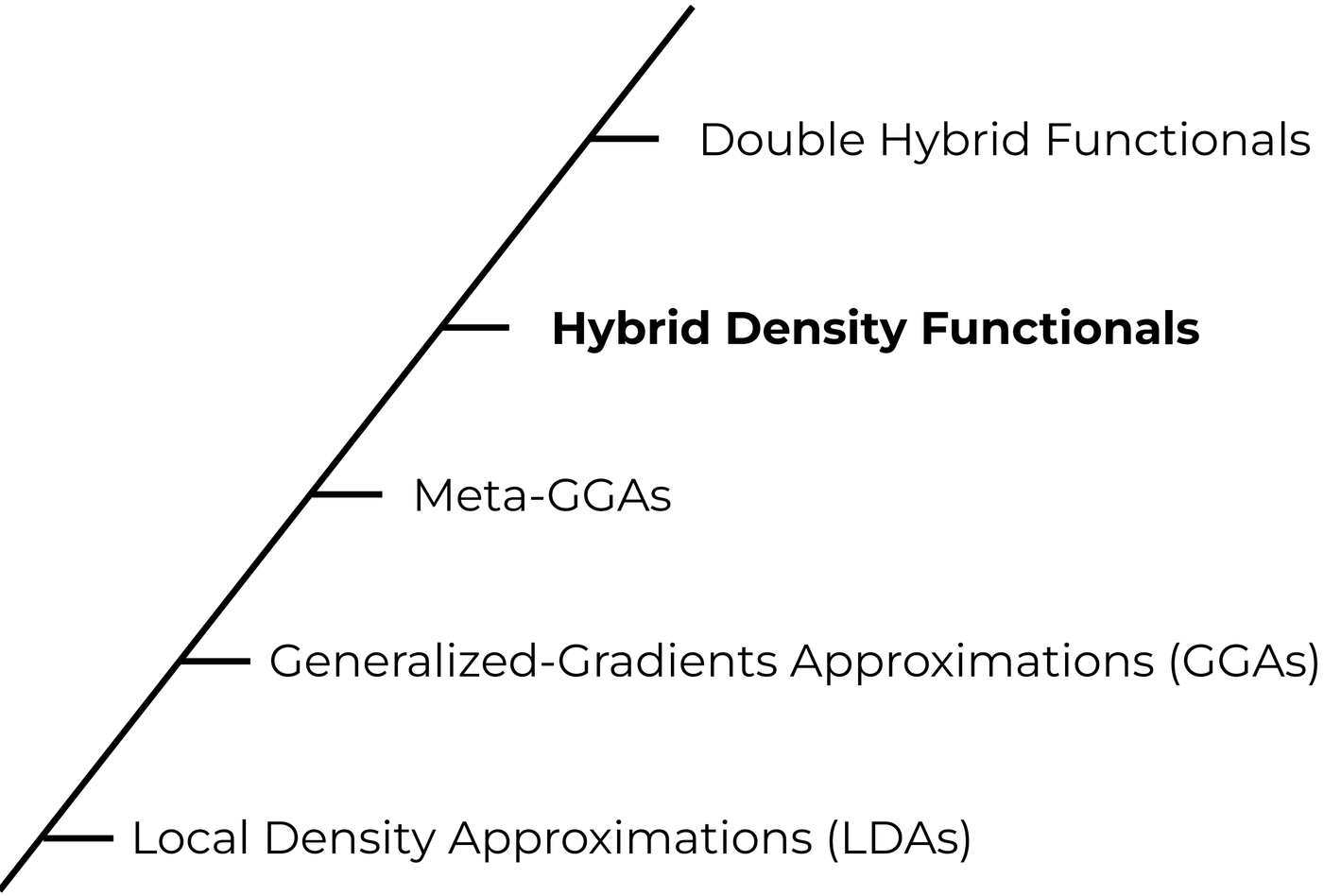
1. Focus Talk
2. Hands-on time – Feel free to ask questions, also unrelated

PART A

Hybrid Density Functionals

Density Functional Approximations (DFAs)

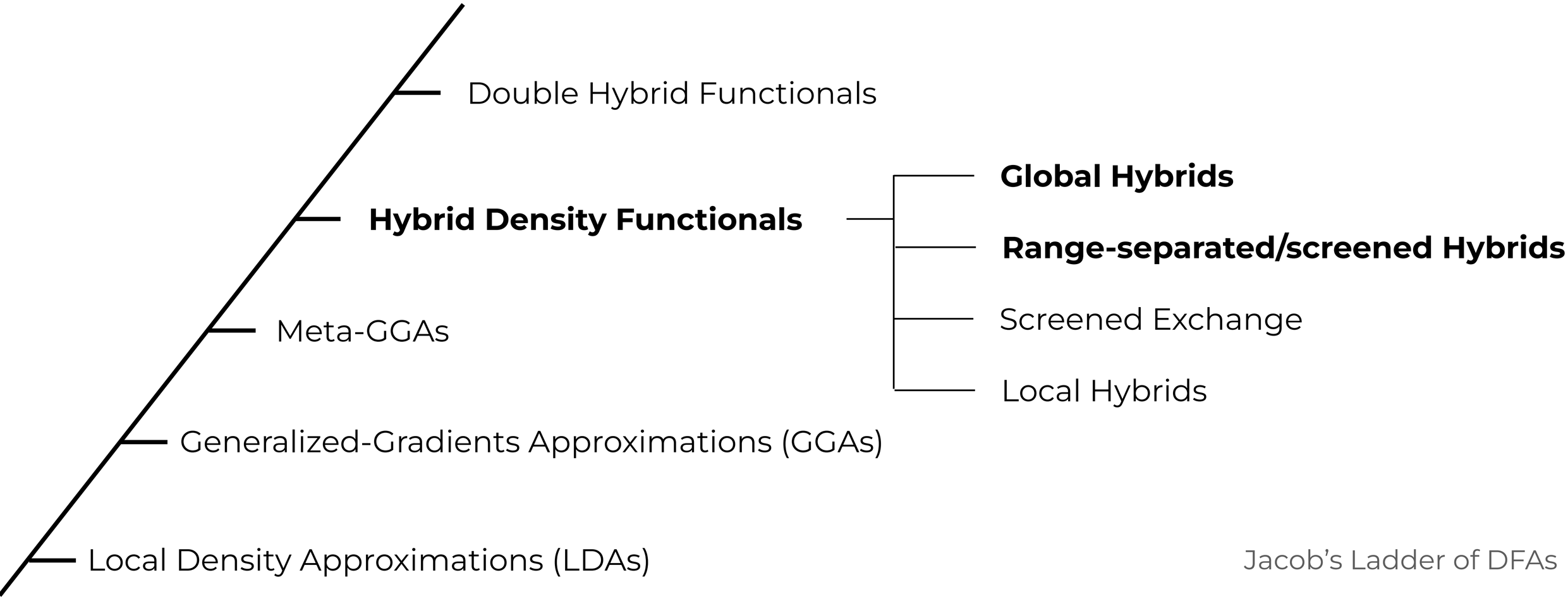
At which rung we stop today



Jacob’s Ladder of DFAs

Density Functional Approximations (DFAs)

At which rung we stop today



Jacob's Ladder of DFAs

How to Construct Hybrid Functionals

Global and range-separated/screened Hybrids

How to Construct Hybrid Functionals

Global and range-separated/screened Hybrids

Mixing the (screened) DFA-exchange with some fraction of non-local Exact Exchange (exx)

$$E_x(\alpha, \beta, \omega) = \alpha E_{\mathrm{exx}} + \beta E^{\mathrm{SR}}_{\mathrm{exx}}(\omega) + (1-\alpha) E_{\text{x-DFA}} - \beta E^{\mathrm{SR}}_{\text{x-DFA}}(\omega)$$

How to Construct Hybrid Functionals
Global and range-separated/screened Hybrids

Mixing the (screened) DFA-exchange with some fraction of non-local Exact Exchange (exx)

$$E_x(\alpha, \beta, \omega) = \alpha E_{\text{exx}} + \beta E^{\text{SR}}_{\text{exx}}(\omega) + (1-\alpha) E_{\text{x-DFA}} - \beta E^{\text{SR}}_{\text{x-DFA}}(\omega)$$

β = 0: Global Hybrids

PBE0, PBEsol0, B3LYP, …

How to Construct Hybrid Functionals

Global and range-separated/screened Hybrids

Mixing the (screened) DFA-exchange with some fraction of non-local Exact Exchange (exx)

$$E_x(\alpha, \beta, \omega) = \alpha E_{\text{exx}} + \beta E^{\text{SR}}_{\text{exx}}(\omega) + (1-\alpha) E_{\text{x-DFA}} - \beta E^{\text{SR}}_{\text{x-DFA}}(\omega)$$

β = 0: Global Hybrids

PBE0, PBEsol0, B3LYP, …

β > 0: Range-separated/screened Hybrids

HSE06, ωPBEh, ωB97h, …

How to Construct Hybrid Functionals

Global and range-separated/screened Hybrids

Mixing the (screened) DFA-exchange with some fraction of non-local Exact Exchange (exx)

$$E_x(\alpha, \beta, \omega) = \alpha E_{\text{exx}} + \beta E^{\text{SR}}_{\text{exx}}(\omega) + (1-\alpha) E_{\text{x-DFA}} - \beta E^{\text{SR}}_{\text{x-DFA}}(\omega)$$

β = 0: Global Hybrids

PBE0, PBEsol0, B3LYP, …

β > 0: Range-separated/screened Hybrids

HSE06, ωPBEh, ωB97h, …

α – long-range exx

α+β – short-range exx

How to Construct Hybrid Functionals

Global and range-separated/screened Hybrids

Mixing the (screened) DFA-exchange with some fraction of non-local Exact Exchange (exx)

$$E_x(\alpha, \beta, \omega) = \alpha E_{\mathrm{exx}} + \beta E^{\mathrm{SR}}_{\mathrm{exx}}(\omega) + (1-\alpha) E_{\text{x-DFA}} - \beta E^{\mathrm{SR}}_{\text{x-DFA}}(\omega)$$

SHORT-RANGE LONG-RANGE

$$v(r) = v_{\mathrm{SR}}(r;\omega) + v_{\mathrm{LR}}(r;\omega) = \frac{\mathrm{erfc}(\omega r)}{r} + \frac{\mathrm{erf}(\omega r)}{r}$$

α – long-range exx

α+β – short-range exx

How to Construct Hybrid Functionals

Global and range-separated/screened Hybrids

Mixing the (screened) DFA-exchange with some fraction of non-local Exact Exchange (exx)

$$E_x(\alpha, \beta, \omega) = \alpha E_{\text{exx}} + \beta E^{\text{SR}}_{\text{exx}}(\omega) + (1-\alpha) E_{\text{x-DFA}} - \beta E^{\text{SR}}_{\text{x-DFA}}(\omega)$$

SHORT-RANGE LONG-RANGE

$$v(r) = v_{\text{SR}}(r;\omega) + v_{\text{LR}}(r;\omega) = \frac{\text{erfc}(\omega r)}{r} + \frac{\text{erf}(\omega r)}{r}$$

Hybrid Functionals are a very flexible scheme!

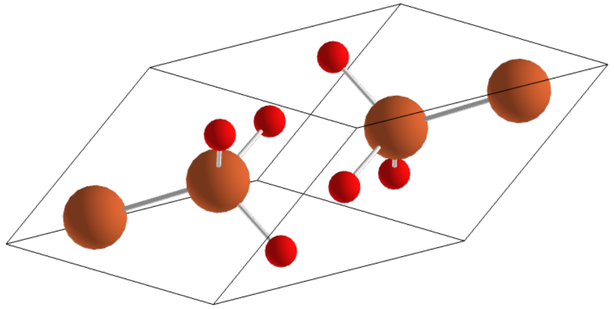
(That's good/bad)

α – long-range exx

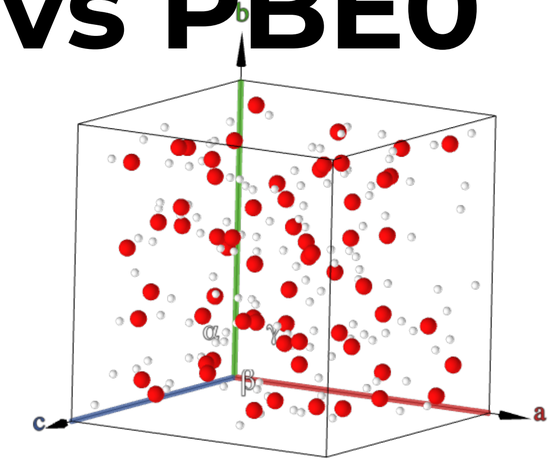
α+β – short-range exx

Practical Guide for Using Hybrids

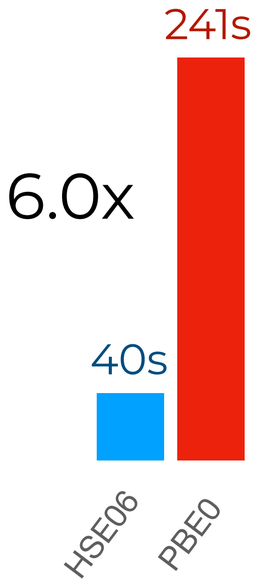
Hybrid ≠ Hybrid: HSE06 vs PBE0



Hematite Fe_2O_3, anti-ferromagnetic

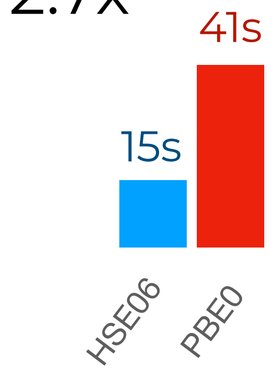


64 water molecules in box



Run on 2 nodes á 72 cores

Similar behavior for memory consumption



Run on 5 nodes á 72 cores

Efficient evaluation of exact exchange in real space

Localized Resolution-of-Identity approach (RI-LVL) [1,2]

Exchange matrix

Density matrix

$$X_{ij}^{\sigma}(\mathbf{R}) = \sum_{kl}\sum_{\mathbf{R}'} D_{kl}^{\sigma}(\mathbf{R}')$$

Basis functions

$$\times \sum_{\mathbf{R}''}\int d\mathbf{r}d\mathbf{r}' \frac{\phi_i(\mathbf{r})\phi_k(\mathbf{r}-\mathbf{R}'')\phi_j(\mathbf{r}'-\mathbf{R})\phi_l(\mathbf{r}'-\mathbf{R}'-\mathbf{R}'')}{|\mathbf{r}-\mathbf{r}'|}$$

4-center-2-electrons integral (4 indices)

[1] Ren, X., et al. New Journal of Physics 14.5 (2012): 053020.
[2] Ihrig, A., *et al.* New Journal of Physics 17.9 (2015): 093020.

Efficient evaluation of exact exchange in real space

Localized Resolution-of-Identity approach (RI-LVL) [1,2]

Exchange matrix

Density matrix

$$X_{ij}^{\sigma}(\mathbf{R}) = \sum_{kl}\sum_{\mathbf{R}'} D_{kl}^{\sigma}(\mathbf{R}') \times \sum_{\mathbf{R}''}\int d\mathbf{r}d\mathbf{r}' \frac{\phi_i(\mathbf{r})\phi_k(\mathbf{r}-\mathbf{R}'')\phi_j(\mathbf{r}'-\mathbf{R})\phi_l(\mathbf{r}'-\mathbf{R}'-\mathbf{R}'')}{|\mathbf{r}-\mathbf{r}'|}$$

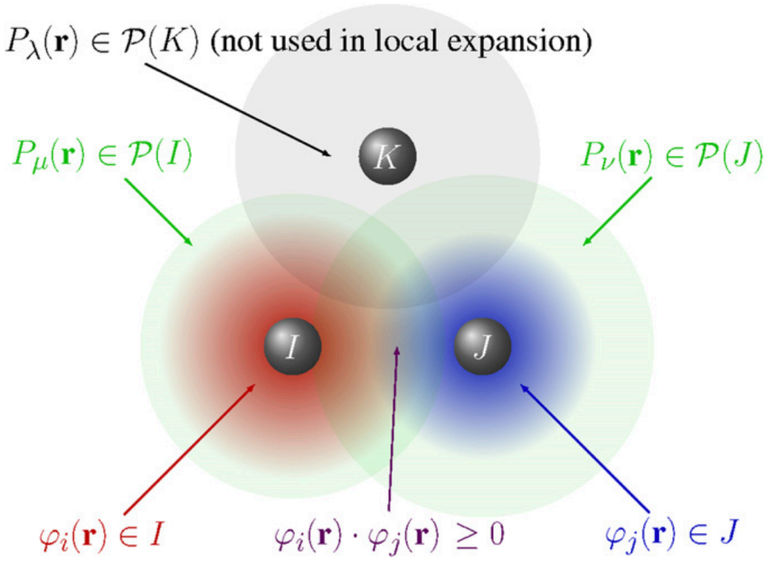
Basis functions

4-center-2-electrons integral (4 indices)

Localized Resolution-of-Identity (RI-LVL) [2]

Auxiliary basis set

$$\phi_i(r)\phi_j(r) := \rho_{ij}(r) \approx \sum_{\mu\in\mathscr{P}(IJ)} C_{ij}^{\mu} P_{\mu}(r)$$



[1] Ren, X., et al. New Journal of Physics 14.5 (2012): 053020.
[2] Ihrig, A., *et al.* New Journal of Physics 17.9 (2015): 093020.

Efficient evaluation of exact exchange in real space

Localized Resolution-of-Identity approach (RI-LVL) [1,2]

Exchange matrix

Density matrix

$$X_{ij}^{\sigma}(\mathbf{R}) = \sum_{kl}\sum_{\mathbf{R}'} D_{kl}^{\sigma}(\mathbf{R}')$$

Basis functions

$$\times \sum_{\mathbf{R}''}\int d\mathbf{r}d\mathbf{r}' \frac{\phi_i(\mathbf{r})\phi_k(\mathbf{r}-\mathbf{R}'')\phi_j(\mathbf{r}'-\mathbf{R})\phi_l(\mathbf{r}'-\mathbf{R}'-\mathbf{R}'')}{|\mathbf{r}-\mathbf{r}'|}$$

4-center-2-electrons integral (4 indices)

RI-LVL

Coulomb Matrix (2 indices)

$$\mathbf{X} = \mathbf{C}\cdot\mathbf{V}\cdot\mathbf{C}'\cdot\mathbf{D}$$

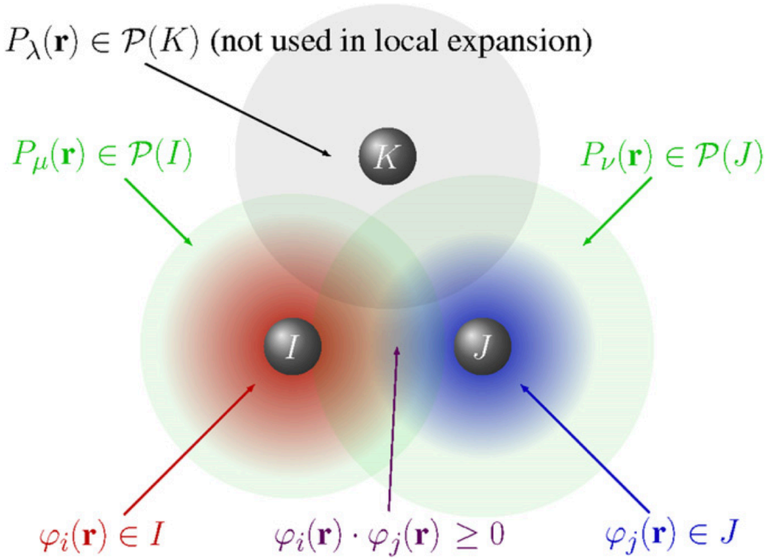
RI expansion coefficients (3 indices)

[1] Ren, X., et al. New Journal of Physics 14.5 (2012): 053020.
[2] Ihrig, A., *et al.* New Journal of Physics 17.9 (2015): 093020.

Localized Resolution-of-Identity (RI-LVL) [2]

Auxiliary basis set

$$\phi_i(r)\phi_j(r) := \rho_{ij}(r) \approx \sum_{\mu\in\mathscr{P}(IJ)} C_{ij}^{\mu} P_{\mu}(r)$$



RI-LVL enables O(n_atoms)

Exchange matrix

Density matrix

$$X_{ij}^{\sigma}(\mathbf{R}) = \sum_{kl}\sum_{\mathbf{R}'} D_{kl}^{\sigma}(\mathbf{R}')$$

Basis functions

$$\times \sum_{\mathbf{R}''}\int d\mathbf{r}d\mathbf{r}' \frac{\phi_i(\mathbf{r})\phi_k(\mathbf{r}-\mathbf{R}'')\phi_j(\mathbf{r}'-\mathbf{R})\phi_l(\mathbf{r}'-\mathbf{R}'-\mathbf{R}'')}{|\mathbf{r}-\mathbf{r}'|}$$

4-center-2-electrons integral (4 indices)

RI-LVL

Coulomb Matrix (2 indices)

$$\mathbf{X} = \mathbf{C}\cdot\mathbf{V}\cdot\mathbf{C}'\cdot\mathbf{D}$$

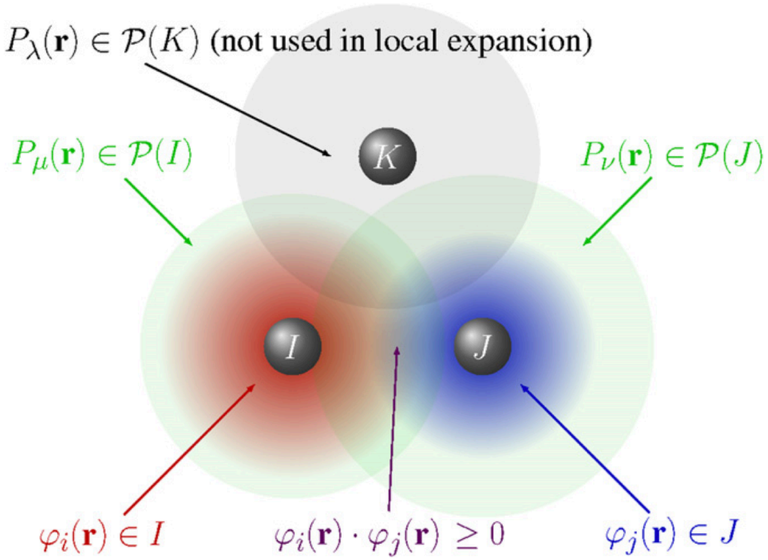
RI expansion coefficients (3 indices)

[1] Ren, X., et al. New Journal of Physics 14.5 (2012): 053020.
[2] Ihrig, A., *et al.* New Journal of Physics 17.9 (2015): 093020.

Localized Resolution-of-Identity (RI-LVL) [2]

Auxiliary basis set

$$\phi_i(r)\phi_j(r) := \rho_{ij}(r) \approx \sum_{\mu\in\mathscr{P}(IJ)} C_{ij}^{\mu} P_{\mu}(r)$$



Hybrid Functionals in FHI-aims

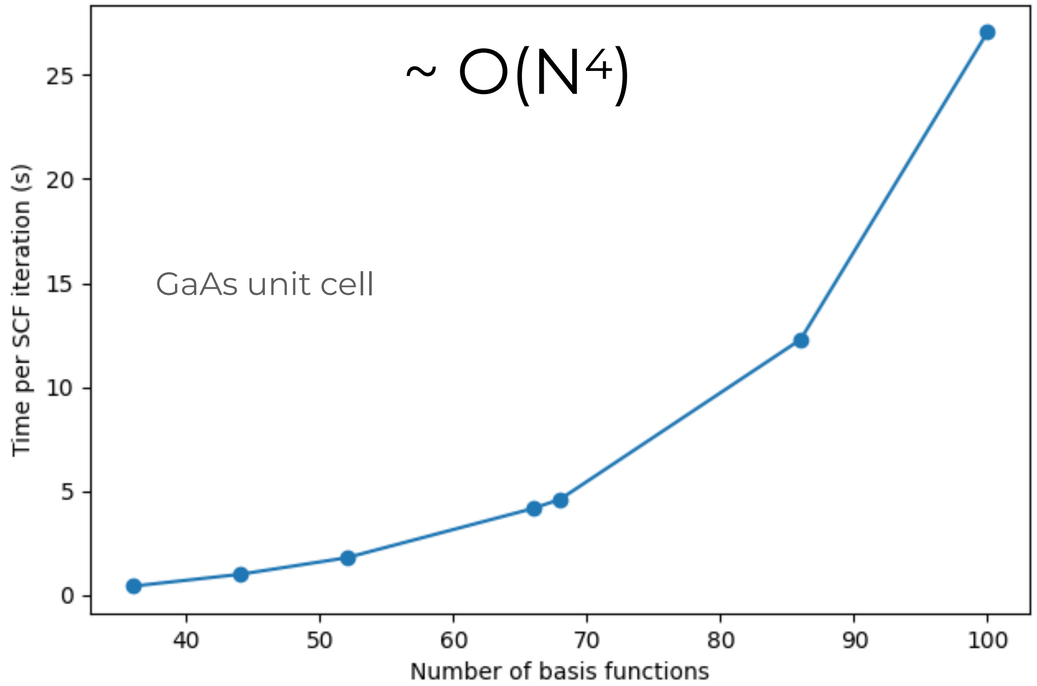
Benefits of the RI-LVL implementation

- Available for non-periodic and periodic structures
- Affordable (still not cheap);
- O(n_atoms) algorithm: Simulations up to ~30k atoms
- Integration with libxc: find you favorite hybrid functional

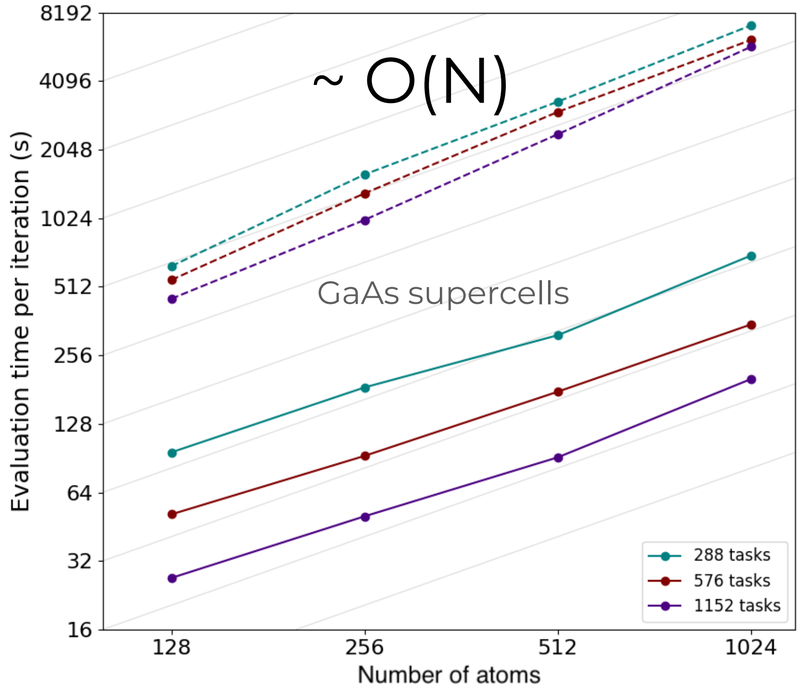
Practical Guide for Using Hybrids

Complexity of the Exact Exchange Algorithm

Number of Basis Functions per Atom



Number of Atoms



Use scaling behavior to estimate computational resources from smaller units

Recent Exact Exchange Code Improvements

CPUs only!

Recent Exact Exchange Code Improvements

CPUs only!

- MPI-3 intra-node shared memory arrays (see talk Christian Carbogno)
  - Facilitate data redistribution via one-sided MPI routines; e.g. Coulomb matrix

Recent Exact Exchange Code Improvements

CPUs only!

- MPI-3 intra-node shared memory arrays (see talk Christian Carbogno)
  - Facilitate data redistribution via one-sided MPI routines; e.g. Coulomb matrix
- Compression of the RI-coefficients
  - Same screening mechanism as for Coulomb matrix (2015)

Recent Exact Exchange Code Improvements

CPUs only!

- MPI-3 intra-node shared memory arrays (see talk Christian Carbogno)
  - Facilitate data redistribution via one-sided MPI routines; e.g. Coulomb matrix
- Compression of the RI-coefficients
  - Same screening mechanism as for Coulomb matrix (2015)
- Several auto-tuning mechanisms
  - Minimize communication, maximize memory/node usage

Recent Exact Exchange Code Improvements

CPUs only!

- MPI-3 intra-node shared memory arrays (see talk Christian Carbogno)
  - Facilitate data redistribution via one-sided MPI routines; e.g. Coulomb matrix
- Compression of the RI-coefficients
  - Same screening mechanism as for Coulomb matrix (2015)
- Several auto-tuning mechanisms
  - Minimize communication, maximize memory/node usage
- Additional parallelization layers
  - Finer granulation on the level of basis functions (previously: on the level of atoms)

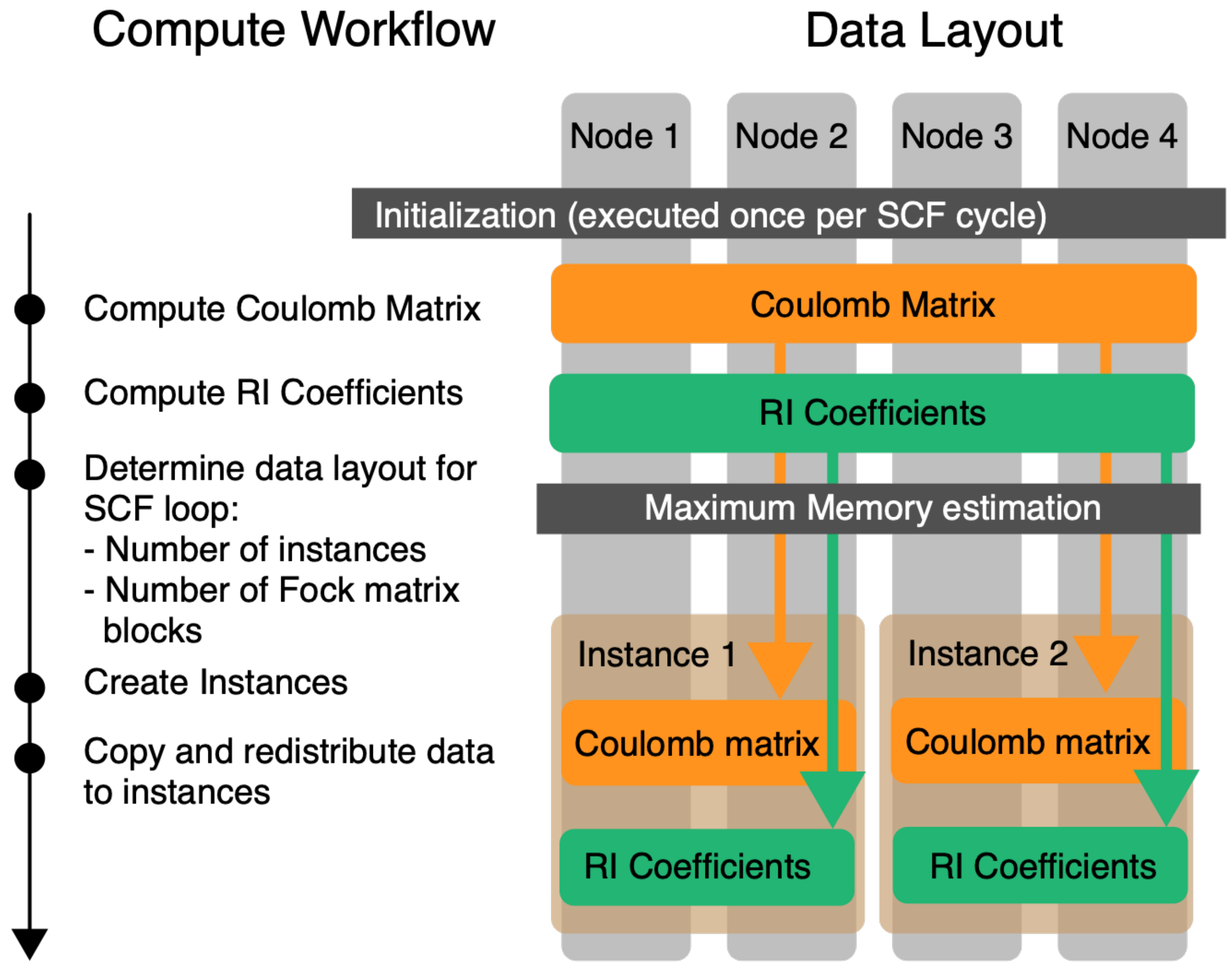
Workflow Overview

Initialization

$$\mathbf{X} = \mathbf{C} \cdot \mathbf{V} \cdot \mathbf{C}' \cdot \mathbf{D}$$

Instance: everything that is needed to compute a row of the Fock matrix X:
- Coulomb Matrix C
- RI Coefficients V
- Communicators
- Several index arrays

The total number of instances needs to Be divisor of the number of nodes



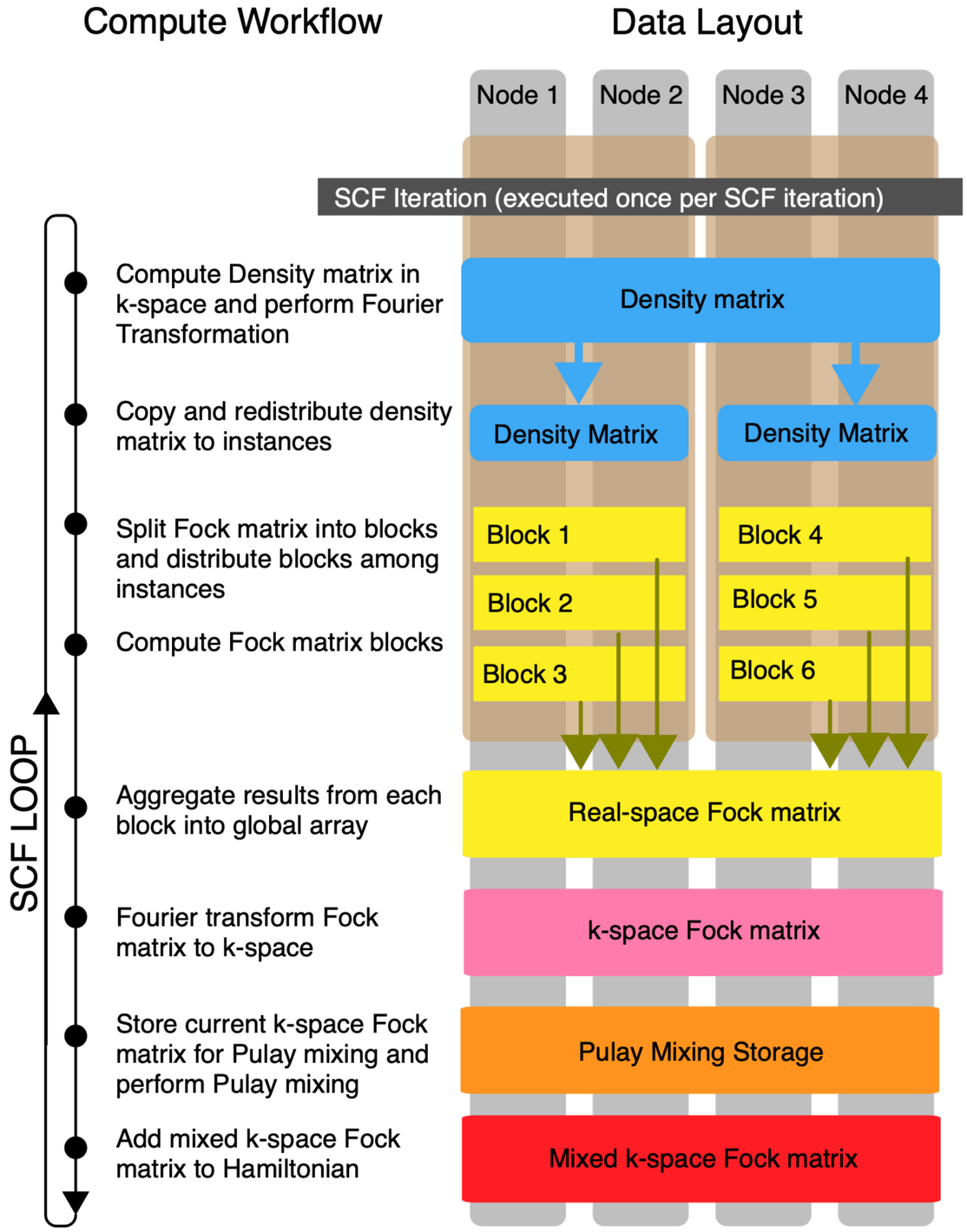
Kokott, Merz, *et al.*, arXiv:2403.10343 (2024)

Workflow Overview

SCF iterations

$$\mathbf{X} = \mathbf{C} \cdot \mathbf{V} \cdot \mathbf{C}' \cdot \mathbf{D}$$

Block: Rows of the Fock matrix X



Kokott, Merz, *et al.*, arXiv:2403.10343 (2024)

Practical Guide for Using Hybrids

Important things to learn from the output

1,503-atom calculation, 1x1x1 k-grid, ran on 16 nodes à 72 MPI tasks

```
Assumed free memory per node for Fock matrix prefactors [GB]    185.59
Memory per instance for Coulomb / ovlp matrix [GB]    252.76    589.20
Number of nodes, tasks per node    16    72
Number of instances:    2
```

Practical Guide for Using Hybrids

Important things to learn from the output

1,503-atom calculation, 1x1x1 k-grid, ran on 16 nodes à 72 MPI tasks

```
Assumed free memory per node for Fock matrix prefactors [GB]    185.59
Memory per instance for Coulomb / ovlp matrix [GB]    252.76    589.20
Number of nodes, tasks per node    16    72
Number of instances:    2
```

How much assumed free memory we have in total: n_nodes x 185.59 GB = 2,969.44 GB

How much memory per instance we need: M_Coulomb + M_OVLP = 841.96 GB

Ratio 2,969.44/841.96 ≈ 3

Practical Guide for Using Hybrids

Important things to learn from the output

1,503-atom calculation, 1x1x1 k-grid, ran on 16 nodes à 72 MPI tasks

```
Assumed free memory per node for Fock matrix prefactors [GB]    185.59
Memory per instance for Coulomb / ovlp matrix [GB]    252.76    589.20
Number of nodes, tasks per node    16    72
Number of instances:    2
```

How much assumed free memory we have in total: n_nodes x 185.59 GB = 2,969.44 GB

How much memory per instance we need: M_Coulomb + M_OVLP = 841.96 GB

Ratio 2,969.44/841.96 ≈ 3

MINIMAL NODE SETUP ESTIMATION

Calculation is likely to run on **8 nodes**

Practical Guide for Using Hybrids

Important things to learn from the output

1,503-atom calculation, 1x1x1 k-grid, ran on 16 nodes à 72 MPI tasks

```
Assumed free memory per node for Fock matrix prefactors [GB]    185.59
Memory per instance for Coulomb / ovlp matrix [GB]    252.76    589.20
Number of nodes, tasks per node    16    72
Number of instances:    2
```

How much assumed free memory we have in total: n_nodes x 185.59 GB = 2,969.44 GB

How much memory per instance we need: M_Coulomb + M_OVLP = 841.96 GB

Ratio 2,969.44/841.96 ≈ 3

MINIMAL NODE SETUP ESTIMATION

Calculation is likely to run on **8 nodes**

BEST SETUP IN A SIMILAR NODE RANGE

15 nodes will allow to open 3 instances

Practical Guide for Using Hybrids

How to tune your Hybrid Functional Simulation

FHI-aims tries to auto-tune your setup

- Number of instances, Fock matrix block size, Number of Fock matrix blocks per node

Defaults are chosen rather conservatively

- Try to avoid out-of-memory runs

Possibly gain another ~20% speed-up when manually tuned

Practical Guide for Using Hybrids

How to tune your Hybrid Functional Simulation

Set the number of instances per node

```
fock_matrix_nodes_per_instance  INTEGER
! INTEGER has to be a divisor of number of nodes
```

Set the Fock matrix block size

```
fock_matrix_blocking  INTEGER
! INTEGER should be smaller than 100
```

Limit the number of available memory per node

```
fock_matrix_max_mem_per_node  FLOAT
```

Efficient Workflows for Hybrid functionals

(May not always apply to your system)

1. GGA relaxation with light species defaults
2. Follow-up relaxation with Hybrid Functional (forces + stress) with intermediate defaults
3. Band structure calculation with intermediate species defaults

Try it yourself with Tutorial
Basics Of Running FHI-aims: Part 3

Where to find the material

Go to fhi-aims.org

FHI-aims version (special release)

➡️ Download fhi-aims.240507 or
git checkout 240507

➡️ Compile FHI-aims

Online Tutorial Material

➡️ Click Online tutorials

➡️ Select: Basics of Running FHI-aims

- Part 3

PART B

GW method for molecules and solids