

Introduction to BigDFT

Dr. Laura Ratcliff

School of Chemistry, University of Bristol
UiT The Arctic University of Norway

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Outline

① What are Wavelets?

[Haar Wavelets](#)[Daubechies Wavelets](#)

② BigDFT

[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)

③ Large Systems

[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)

④ In Practice

⑤ Summary

Outline

① What are Wavelets?

Haar Wavelets

Daubechies Wavelets

② BigDFT

DFT with Daubechies Wavelets

BigDFT in Practice

③ Large Systems

The Formalism

LS-BigDFT in Action

Fragments in BigDFT

④ In Practice

⑤ Summary

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Advantages of Wavelets

Many alternative basis sets, but ideally want to combine the advantages of Gaussians with those of plane waves → what does that look like?

Wavelets are used in other fields e.g. signal processing, but are uncommon for electronic structure simulations

Wavelets come in families, each with their own features, but have in common some key properties:

- localised in real and Fourier space ✓
- systematic convergence ✓
- adaptivity ✓

→ potential advantages for DFT

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Introducing Wavelets I

Wavelet families are characterised by two mother functions:

- the scaling function $\phi(x)$
- the wavelet $\psi(x)$

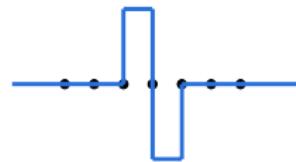
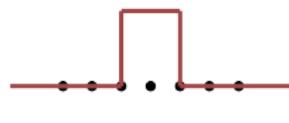
We can use the following relations to dilate (j) and translate (k) the basis functions:

$$\phi_j^k(x) = 2^{j/2} \phi(2^j x - k); \quad \psi_j^k(x) = 2^{j/2} \psi(2^j x - k)$$

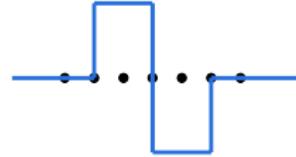
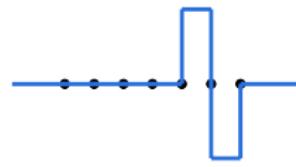
→ the full basis is generated by translating $\phi(x)$ and $\psi(x)$ across a grid with spacing h

Introducing Wavelets II

Let's take a look at the simplest wavelet family: Haar wavelets



Translation and dilation:



→ flexibility in how to represent an arbitrary function:

- using a purely **scaling function** based representation
- using a mix of **scaling functions** and **wavelets**

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Introducing Wavelets III

Relationship between basis functions → important feature of wavelets

E.g., we can relate a **scaling function** at a given resolution (grid spacing h) to **scaling functions** at twice the resolution (i.e. $h/2$):

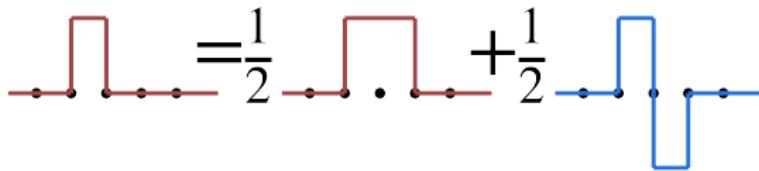


Similarly, we can relate a **wavelet** at grid spacing h to **scaling functions** at grid spacing $h/2$:



Introducing Wavelets IV

And can define a **scaling function** at grid spacing $h/2$ in terms of a **scaling function** and a **wavelet** at grid spacing h :



→ can work at twice the resolution ($h/2$) while only using functions defined at resolution h

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Multiresolution/Adaptivity

Can formalise these relationships in the form of the **refinement relations** (aka scaling equations), which relate a wavelet basis set with grid spacing h to one with $h/2$:

$$\phi(x) = \sqrt{2} \sum_j h_j \phi(2x - j)$$

$$\psi(x) = \sqrt{2} \sum_j g_j \phi(2x - j)$$

The coefficients h_j and $g_j = (-1)^j h_{-j+1}$ are the low- and high-pass filters which define the wavelet family

The ability to convert between grids with different resolutions → a **multiresolution** (adaptive) approach. This is a key feature of wavelet basis sets!

Daubechies Wavelets I

Let's look at another type of wavelets – Daubechies wavelets

- they form an orthogonal basis:

$$\langle \phi_i | \phi_j \rangle = \delta_{ij} = \langle \psi_i | \psi_j \rangle; \quad \langle \phi_i | \psi_j \rangle = 0$$

- they have compact support, i.e. are zero outside a given range
- they are relatively smooth
- they have the maximum number of vanishing moments for a given support:
 - Daubechies- $2m$ wavelets have m vanishing moments → can represent any polynomial of degree $< m$ exactly using scaling functions of order m
 - higher order polynomials error $\mathcal{O}(h^m)$ → vanishingly small for small grids

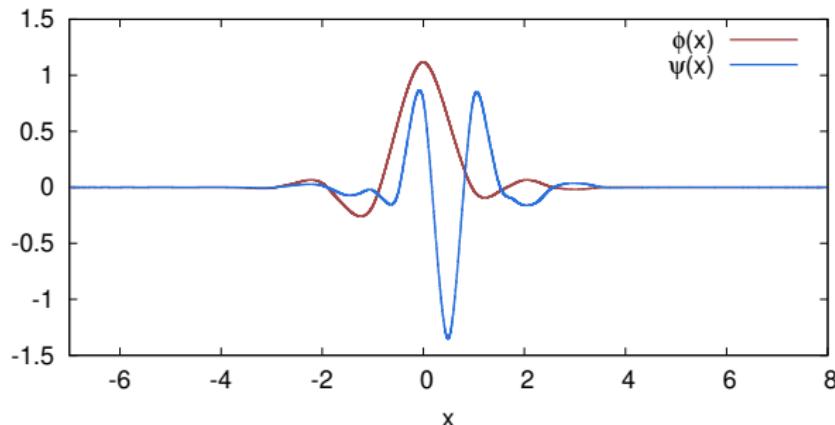
$$\int x^\alpha \psi(x) dx = 0; \quad \alpha = 0, 1 \dots m-1$$

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Daubechies Wavelets II

Example – the least asymmetric Daubechies wavelets of order $2m = 16$:

- compact support within the range $[1 - m, m]$ – localised in real space
- smooth – localised in Fourier space
- polynomial exactness of degree 8



What are
Wavelets?

Haar Wavelets

Daubechies
Wavelets

BigDFT

DFT with
Daubechies
Wavelets

BigDFT in Practice

Large Systems

The Formalism

LS-BigDFT in Action

Fragments in
BigDFT

In Practice

Summary

Outline

1 What are Wavelets?

Haar Wavelets

Daubechies Wavelets

2 BigDFT

DFT with Daubechies Wavelets

BigDFT in Practice

3 Large Systems

The Formalism

LS-BigDFT in Action

Fragments in BigDFT

4 In Practice

5 Summary

BigDFT: DFT with Daubechies Wavelets

The BigDFT project started in 2005, with the aim of writing a DFT code using Daubechies wavelets



Goals:

- efficient treatment of inhomogenous systems (key wavelet features – real space, adaptivity...)
- linear scaling approach for treating 1000s of atoms (key wavelet feature – localisation...)

Key design choices:

- use least asymmetric Daubechies wavelets of order 16
- pseudopotential code – avoids the need for very fine grids
- fixed rather than automatic refinement – 2 levels of resolution only (see MADNESS, MRChem for automatic refinement with multiwavelets)
- range of boundary conditions – free, wire, surface and (orthorhombic) periodic
- open source

Wavelets in BigDFT I

Need to go from a 1D basis to a 3D system – let's define a 3D basis as a tensor product of 1D functions, e.g. for a cubic grid with spacing h' :

$$\phi_{i,j,k}(x, y, z) = \phi(x/h' - i)\phi(y/h' - j)\phi(z/h' - k)$$

Scaling functions and wavelets on a grid of spacing h can be expressed in terms of scaling functions at a finer grid level $h' = h/2 \rightarrow$ can define an equivalent representation with 1 scaling function and 7 wavelets:

$$\phi_{i,j,k}(x, y, z) = \phi(x/h - i)\phi(y/h - j)\phi(z/h - k)$$

$$\psi_{i,j,k}^1(x, y, z) = \psi(x/h - i)\phi(y/h - j)\phi(z/h - k)$$

$$\psi_{i,j,k}^2(x, y, z) = \phi(x/h - i)\psi(y/h - j)\phi(z/h - k)$$

$$\psi_{i,j,k}^3(x, y, z) = \psi(x/h - i)\psi(y/h - j)\phi(z/h - k)$$

$$\psi_{i,j,k}^4(x, y, z) = \phi(x/h - i)\phi(y/h - j)\psi(z/h - k)$$

$$\psi_{i,j,k}^5(x, y, z) = \psi(x/h - i)\phi(y/h - j)\psi(z/h - k)$$

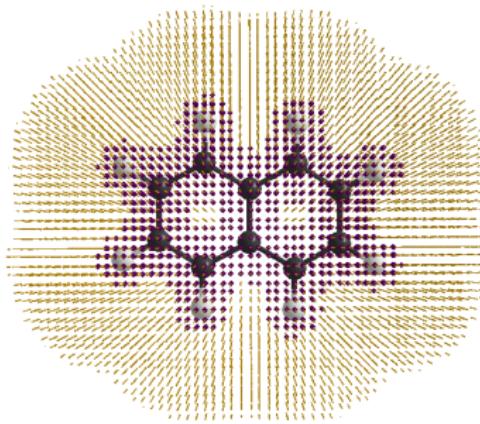
$$\psi_{i,j,k}^6(x, y, z) = \phi(x/h - i)\psi(y/h - j)\psi(z/h - k)$$

$$\psi_{i,j,k}^7(x, y, z) = \psi(x/h - i)\psi(y/h - j)\psi(z/h - k)$$

Wavelets in BigDFT II

Therefore have different regions in BigDFT:

- **fine** – one (3D) **scaling function** and 7 (3D) **wavelets**
- **coarse** – one **scaling function** and no **wavelets** – equivalent to half the resolution of the **fine** region
- **empty space** – no basis functions



The regions are the unions of spheres around each atom a , with:

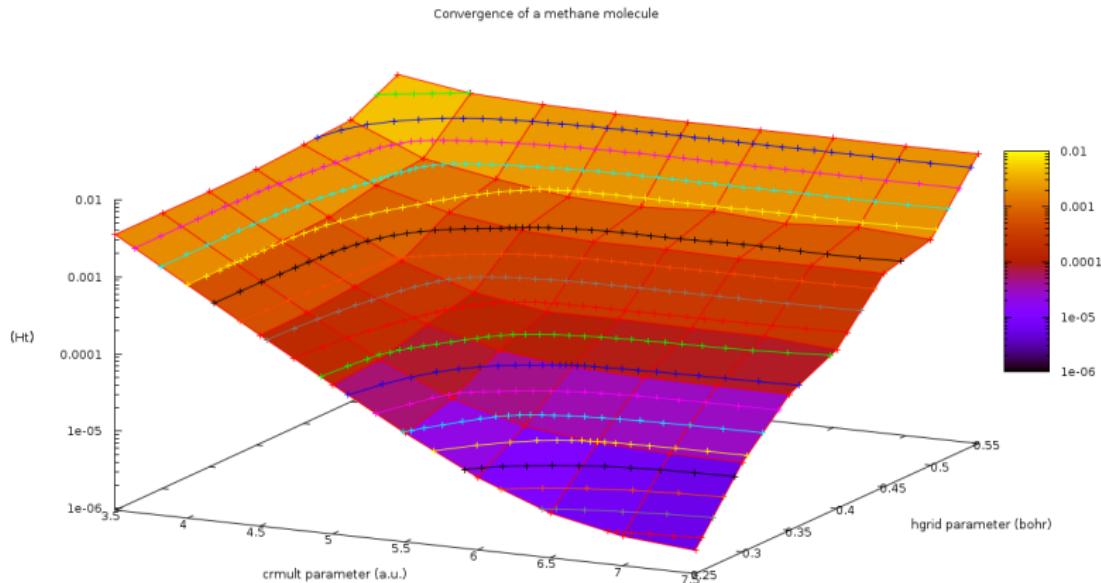
$$R_a^f = \lambda^f r_a^f; \quad R_a^c = \lambda^c r_a^c; \quad (R_a^c > R_a^f)$$

where $r_a^{f(c)}$ are species-dependent values and $\lambda^{f(c)}$ are user-specified

Systematic Convergence I

Systematic convergence with respect to 3 parameters – more basis functions → greater precision

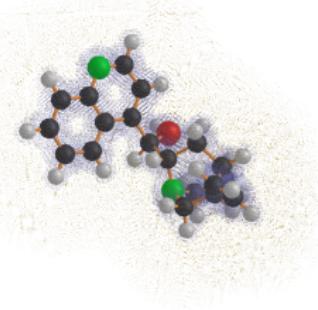
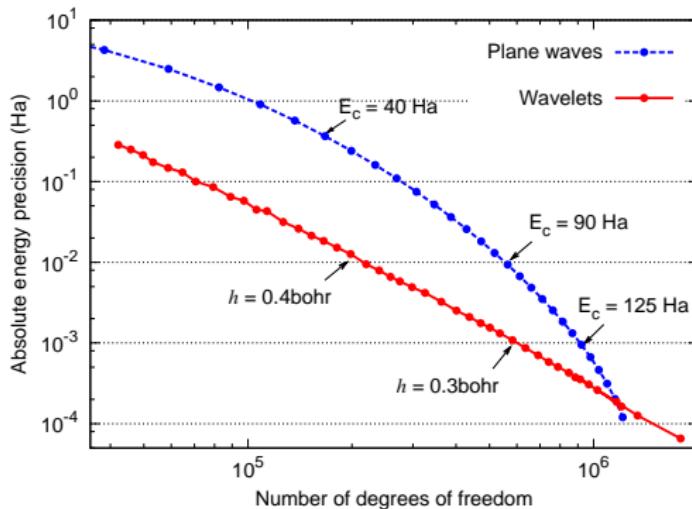
- ↓ grid spacing h – accuracy $\mathcal{O}(h^{14})$ (`hgrid`)
- ↑ λ^c (`crmult`)
- ↑ λ^f (`frmult`)



Systematic Convergence II

Comparison with plane waves (ABINIT):

- algebraic convergence with respect to number of degrees of freedom
- faster with wavelets than plane waves: e.g. cinchonidine



→ advantageous for inhomogeneous systems

Convergence: Genovese, Neelov, Goedecker, Deutsch, Ghasemi, Willand, Caliste, Zilberberg, Rayson, Bergman & Schneider, J. Chem. Phys. **129**, 014109 (2008)

Pseudopotential Accuracy

HGH/GTH pseudopotentials – analytic and separable

Add non-linear core corrections (NLCC):

- close to all electron accuracy, e.g. DeltaTest for elements up to Ar:
- NLCC are relatively soft → can use larger h

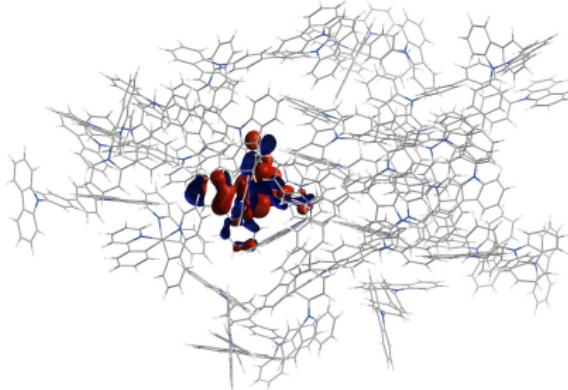
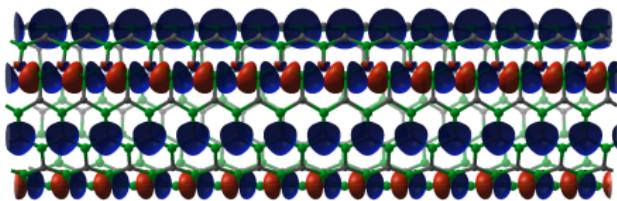
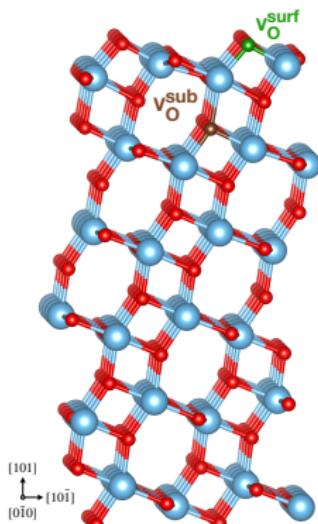
Code	Version	Basis	Electron treatment	Δ -value	Authors
WIEN2k	13.1	LAPW/APW+lo	all-electron	0 meV/atom	S. Cottenier [16]
BigDFT	1.7.6	Daubechies wavelets	HGHk-semicore and NLCC 2015 norm-conserving	0.1 meV/atom	BigDFT [11,16]
FLEUR	0.26	LAPW (+lo)	all-electron	0.2 meV/atom	FLEUR [9,16]
ABINIT	7.8.2	plane waves	PAW JTH v1.0	0.2 meV/atom	F. Jollet and M. Torrent
Quantum ESPRESSO	5.1	plane waves	SSSP Accuracy (mixed NC/US/PAW potential library)	0.2 meV/atom	QuantumESPRESSO [12,16]

Delta Test: <https://molmod.ugent.be/deltacodesdft>
 Lejaeghere, Bihlmayer, Björkman *et al.*, Science 351, 6280 (2016)

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Flexible Boundary Conditions

- easily switch between boundary conditions – no need to add vacuum
- add explicit charges, electric fields in non-periodic directions...
- compare systems of different dimensionalities in **one formalism**



Functionalities

- hybrid functionals (including Hartree Fock)
- k -points
- implicit solvent (separate Poisson solver library)
- structure searching
- excited states: time-dependent DFT and constrained DFT
- large systems:
 - linear-scaling DFT
 - fragments
- ...

Practicalities:

- modular (library-based) structure
- python-based interface (PyBigDFT)
- remote job submission with remote manager
- MPI, OpenMP (and GPU – cubic scaling only)
- **not all functionalities are cross-compatible**

What are
Wavelets?

Haar Wavelets

Daubechies
Wavelets

BigDFT

DFT with
Daubechies
Wavelets

BigDFT in Practice

Large Systems

The Formalism

LS-BigDFT in Action

Fragments in
BigDFT

In Practice

Summary

Outline

1 What are Wavelets?

Haar Wavelets

Daubechies Wavelets

2 BigDFT

DFT with Daubechies Wavelets

BigDFT in Practice

3 Large Systems

The Formalism

LS-BigDFT in Action

Fragments in BigDFT

4 In Practice

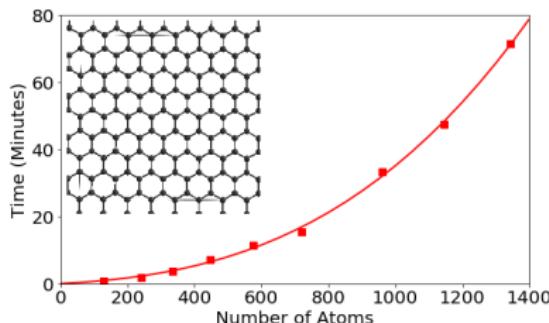
5 Summary

The Cubic Scaling Limit

Scaling with System Size

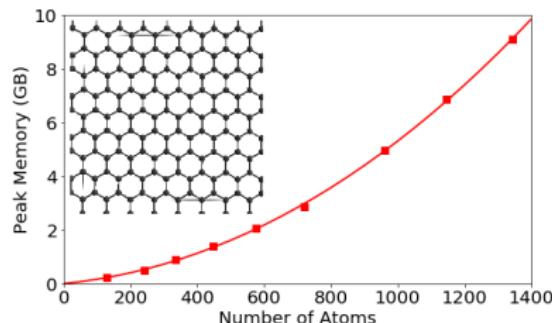
the operations **scale** differently:

- $\mathcal{O}(\mathcal{N} \log \mathcal{N})$: Poisson solver
 - $\mathcal{O}(\mathcal{N}^2)$: convolutions
 - $\mathcal{O}(\mathcal{N}^3)$: linear algebra
- and have different **prefactors**:
- $c_{\mathcal{O}(\mathcal{N}^3)} \ll c_{\mathcal{O}(\mathcal{N}^2)} \ll c_{\mathcal{O}(\mathcal{N} \log \mathcal{N})}$



Size Limitations

- up to ~ 1000 atoms thanks to wavelet properties and efficient parallelisation
- for bigger systems $\mathcal{O}(\mathcal{N}^3)$ dominates
- also reach memory limits
→ need a **new approach**



[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Nearsightedness and Locality

Nearsightedness

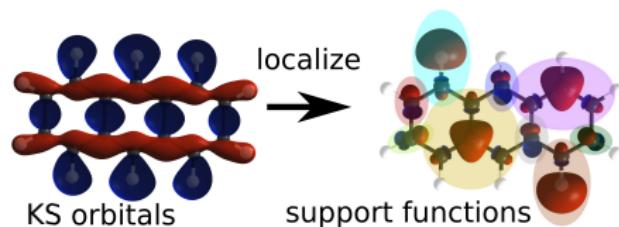
- the behaviour of large systems is **short-ranged** (nearsighted)
 - the density matrix, $\rho(\mathbf{r}, \mathbf{r}')$, decays exponentially in systems with a gap
- how can we exploit nearsightedness to treat large systems?

Support Functions (SFs)

write **extended KS orbitals** in terms of **localised SFs**

$(\{\phi_\alpha(\mathbf{r})\})$:

$$\Psi_i(\mathbf{r}) = \sum_{\alpha} c_i^{\alpha} \phi_{\alpha}(\mathbf{r})$$



Density Matrix Formulation

Avoiding Diagonalisation

- express $\rho(\mathbf{r}, \mathbf{r}')$ in terms of density kernel $K^{\alpha\beta}$ and SFs:

$$\begin{aligned}\rho(\mathbf{r}, \mathbf{r}') &= \sum_i f_i |\Psi_i(\mathbf{r})\rangle\langle\Psi_i(\mathbf{r}')| \\ &= \sum_{\alpha, \beta} |\phi_\alpha(\mathbf{r})\rangle K^{\alpha\beta} \langle\phi_\beta(\mathbf{r}')|\end{aligned}$$

- no explicit reference to KS orbitals – avoid diagonalisation
- orthogonality requirement translates to idempotency:

$$\langle\Psi_i|\Psi_j\rangle = \delta_{ij} \rightarrow \mathbf{K} = \mathbf{KSK}$$

- the density is found via $n(\mathbf{r}) = \rho(\mathbf{r}, \mathbf{r})$

Matrix Expressions

$$H_{\alpha\beta} = \langle\phi_\alpha|\hat{H}|\phi_\beta\rangle; \quad S_{\alpha\beta} = \langle\phi_\alpha|\phi_\beta\rangle$$

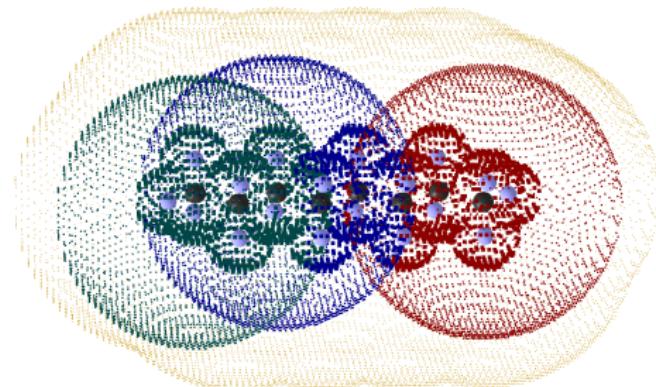
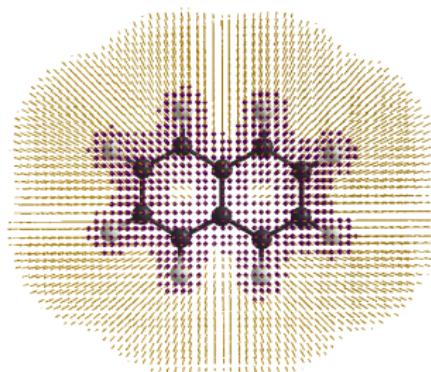
$$E = \text{Tr}(\mathbf{KH}); \quad N = \text{Tr}(\mathbf{KS})$$

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Support Functions

SF Properties

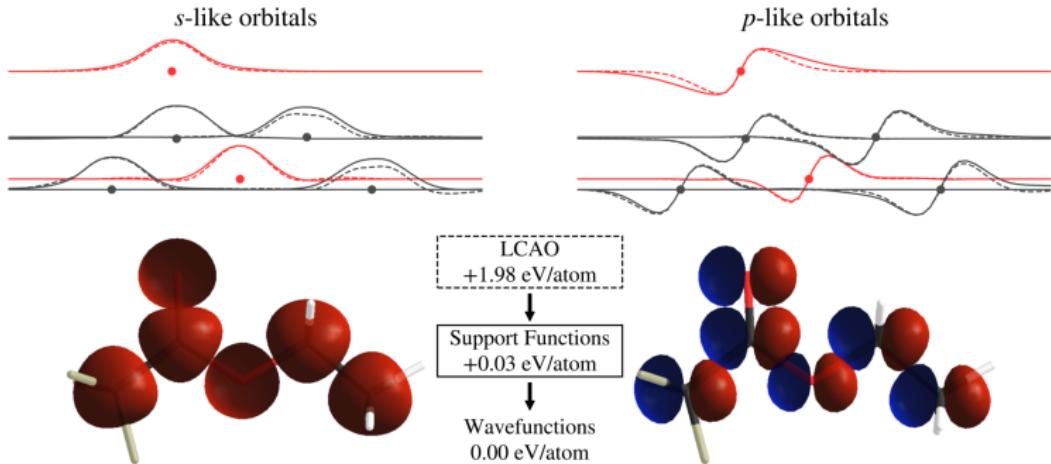
- atom-centred
- strictly localised ($\sim 6 - 8 a_0$ radius) – user defined (systematic convergence)
- numerical functions expanded in wavelets \rightarrow 2 levels of basis ('contracted wavelets')
- (typically) minimal (1 SF per H, 4 per C/N/O...)
- quasi-orthogonal



Support Function Optimisation

How to Have an Accurate Minimal Basis?

- start with atomic orbital guess for the SFs
 - use a gradually decreasing confining potential:
- $$\hat{H}_\alpha = \hat{H} + c_\alpha(\mathbf{r} - \mathbf{R}_\alpha)^4$$
- optimise SFs *in situ* by minimising E (subject to constraints)
 → minimal, localised basis with high accuracy of wavelets



Density Kernel Optimisation

Three Methods for Obtaining K:

- **Diagonalisation**
 - good for small systems and benchmarking/debugging
- **Direct Minimisation**
 - work directly with the KS coefficients
 - can include (low energy) unoccupied states
- **Fermi Operator Expansion (FOE)**
 - density matrix is expressed in terms of \mathbf{H} , i.e. $\mathbf{K} = f(\mathbf{H})$
 - use a Chebyshev polynomial expansion
 - uses a (small) finite temperature – works for metals
 - implemented in CHESS (Chebyshev Sparse Solvers) library

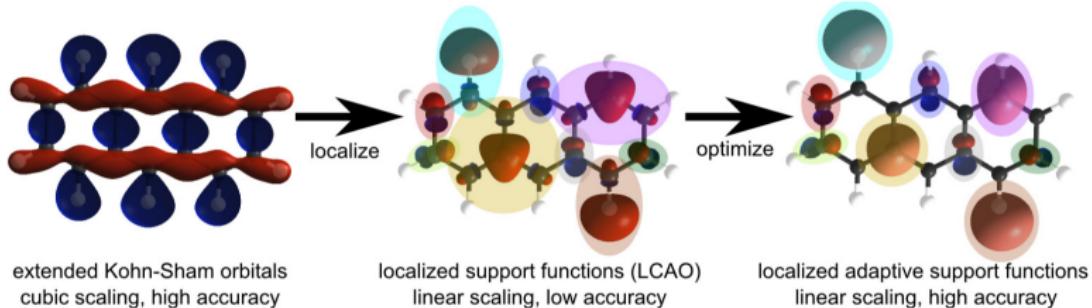
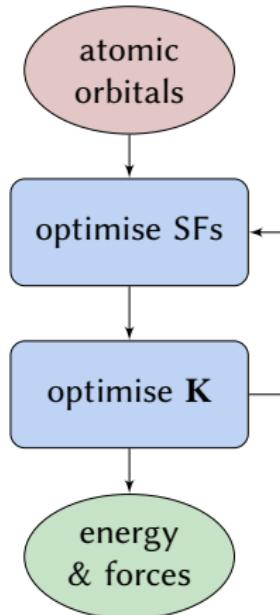
Kernel Truncation

- simple distance criterion $K_{\alpha\beta} = 0$ if $|R_\alpha - R_\beta| > K_{\text{cut}}$

The Algorithm

Calculation Steps

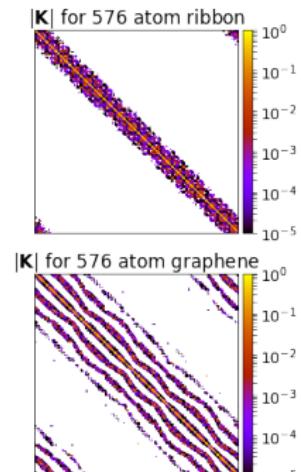
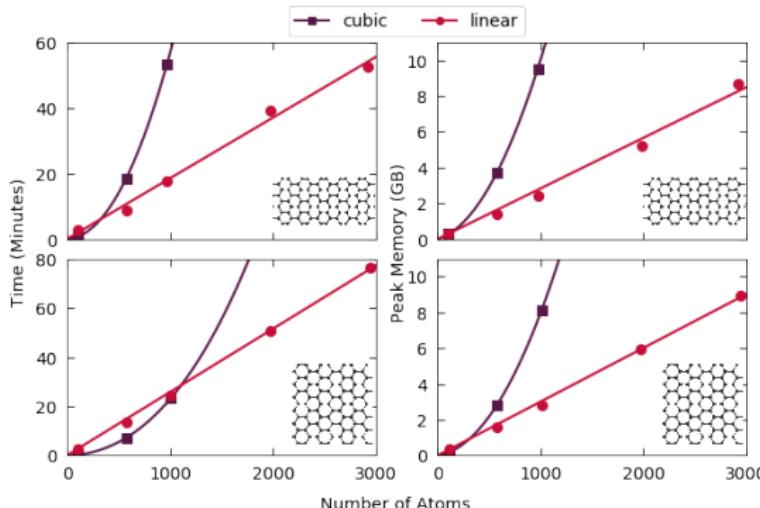
- minimise energy with respect to **both SFs and K**
- **accurate forces** – geometry optimisations, molecular dynamics
- one-off diagonalisation for (occupied) KS energies → densities of states



From Sparsity to Linear Scaling

Sparse Matrices

- strict localisation leads to sparse matrices
- sparsity depends on size, dimensionality, SF radii → crossover
- performance also depends on band gap

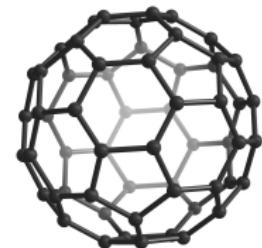
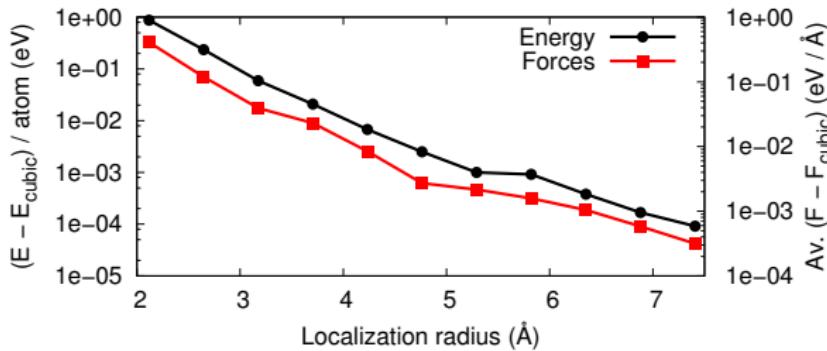


[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Accuracy

Controlling the Accuracy

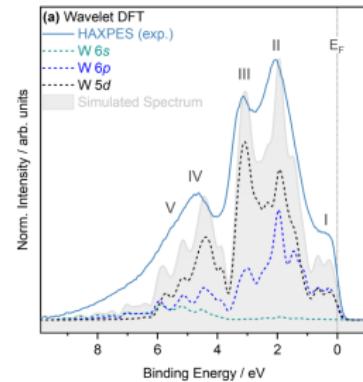
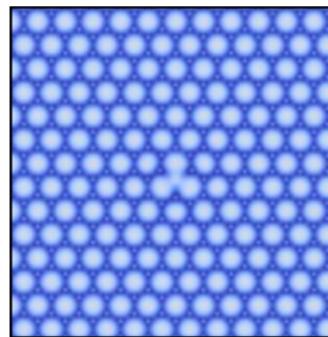
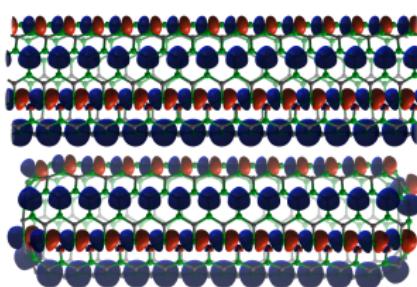
- SF localisation radii are the key parameters
- can choose different values for different atomic species
- can also use **robust profiles**
- energies and forces converge to cubic scaling values



Applicability

Functionalities

- same structure as cubic scaling code – just import a profile
- we don't (can't) use k -points
- same range of boundary conditions
- not all functionalities available in LS-BigDFT: hybrid functionals, TDDFT...



SiC Nanotubes: LER & Genovese, J. Phys.: Condens. Matter **31**, 285901 (2019)

Defective Graphene LER & Genovese, in Theory and Simulation in Physics for Materials Applications, edited by Levchenko, Dappe & Ori, Springer International Publishing (2020)

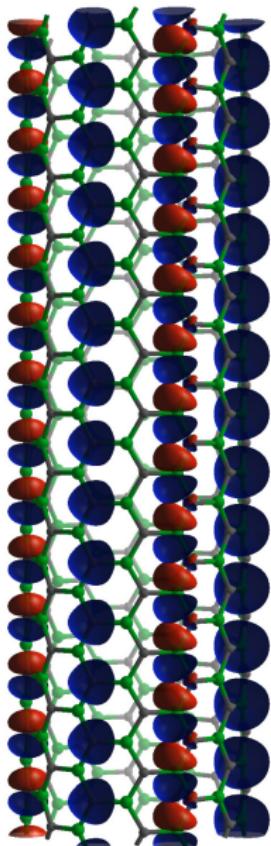
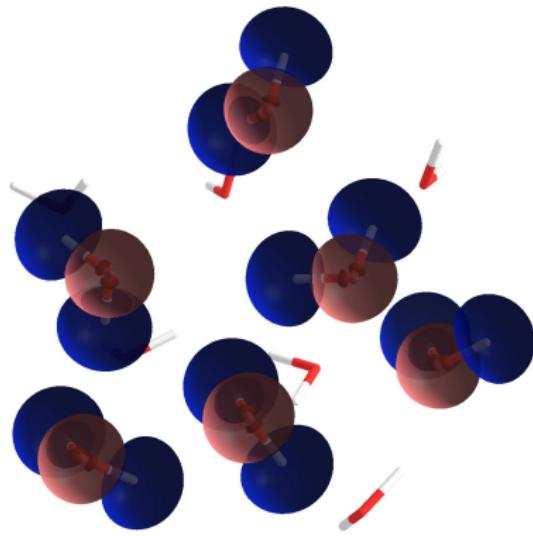
Bulk W: Kalha, LER, Gutiérrez Moreno, Mohr, Mantsinen, Fernando, Thakur, Lee, Tseng, Nunney, Kahk, Lischner & Regoutz, Phys. Rev. B **105**, 045129 (2022)

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

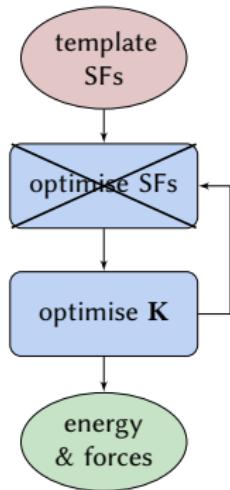
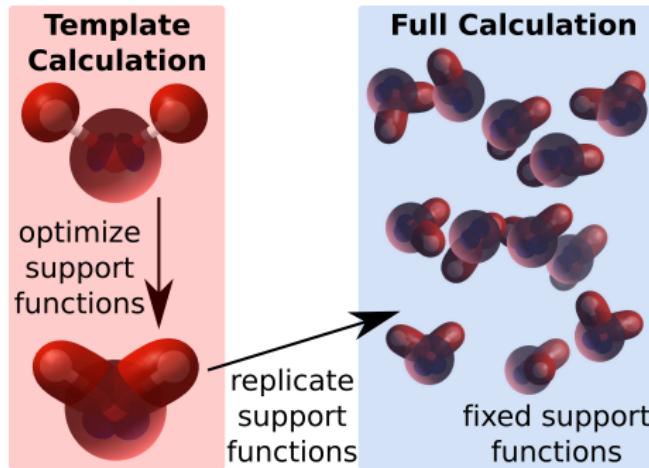
Exploiting Similarity Between Fragments

SF Similarity

- SF optimisation takes a lot of compute time
- what happens in **similar chemical environments**?
- can we reuse SFs?



Molecular Fragment Approach

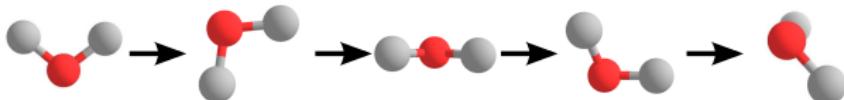


Accounting for Varying Orientations and Positions

- minimise cost function to find **rotation** from template:

$$\mathcal{J}(\mathcal{R}) = \frac{1}{N} \sum_{a=1}^N \left\| \mathbf{R}_a^S - \sum_{b=1}^N \mathcal{R}_{ab} \mathbf{R}_a^T \right\|^2$$

- reformat SFs using **wavelet interpolation**

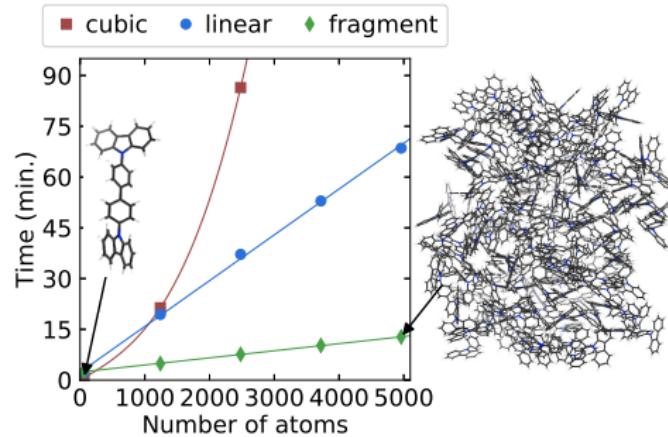
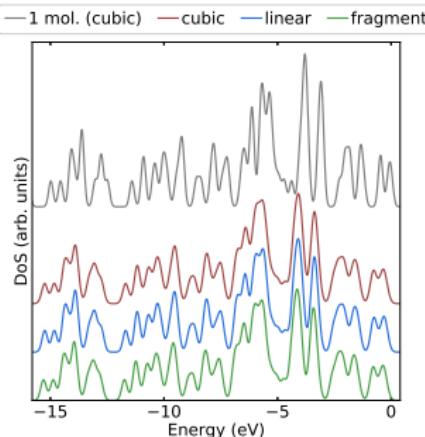


[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Example: Application to OLEDs

Cluster of Rigid CBP Molecules

- ~5000 atom single point calculation (48 nodes on Archer) – fragment approach $\sim 7\times$ cheaper than full linear scaling
- fragment approach reproduces (occupied) DoS
- $E_{\text{frag}} - E_{\text{cubic}} \simeq 30 \text{ meV/atom}$



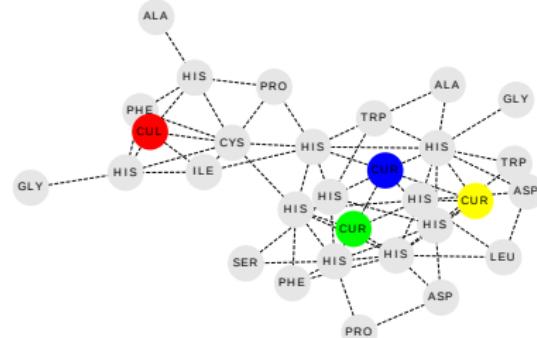
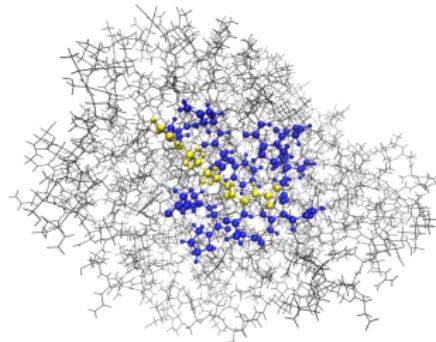
Fragments in BigDFT

Fragment Approach

- weakly interacting fragments which are **not too distorted (\mathcal{J})**
- reduce cost by up to an order of magnitude
- can also apply to **extended systems** using pseudo-fragments

Fragment-Based Analysis: Complexity Reduction

- purity indicator** – identify separable fragments
- fragment bond order** – quantify fragment interactions



Embedded Pseudo-Fragments: LER & Genovese, J. Phys.: Condens. Matter **31**, 285901 (2019)

Complexity Reduction I: Mohr, Masella, LER & Genovese, J. Chem. Theory Comput. **13**, 4079 (2017)

Complexity Reduction II: Dawson, Mohr, LER, Nakajima & Genovese, J. Chem. Theory Comput. **16**, 2952 (2020)

BigDFT: Simulating 1000s of Atoms

What are Wavelets?

Haar Wavelets

Daubechies Wavelets

BigDFT

DFT with Daubechies Wavelets

BigDFT in Practice

Large Systems

The Formalism

LS-BigDFT in Action

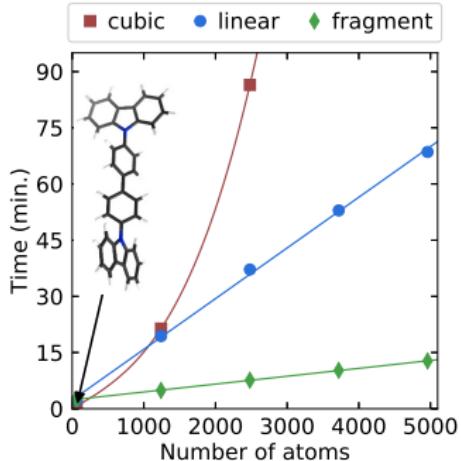
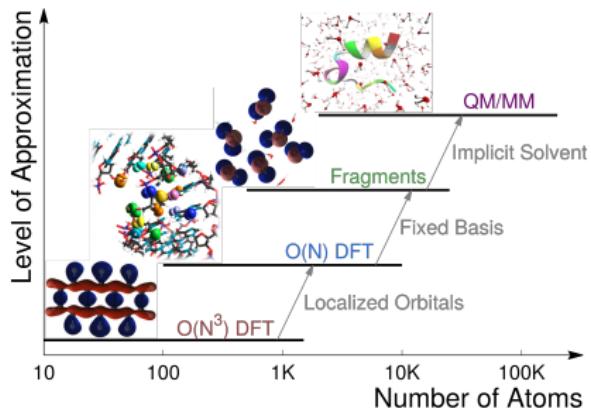
Fragments in BigDFT

In Practice

Summary

BigDFT: From Small to Large Systems

- system size limited to ~ 1000 atoms by **cubic scaling**
- exploit locality with optimized local orbitals \rightarrow **linear scaling**
- exploit repetition \rightarrow **fragment approach** with lower prefactor
- **implicit solvent** \rightarrow Covid spike protein with implicit solvent $\sim 52,000$ atoms in ~ 6 hours on $\sim 16,000$ cores
- can use fragment as a guess for LS, LS as a guess for CS etc.



What are
Wavelets?

Haar Wavelets

Daubechies
Wavelets

BigDFT

DFT with
Daubechies
Wavelets

BigDFT in Practice

Large Systems

The Formalism

LS-BigDFT in Action

Fragments in
BigDFT

In Practice

Summary

Outline

1 What are Wavelets?

Haar Wavelets

Daubechies Wavelets

2 BigDFT

DFT with Daubechies Wavelets

BigDFT in Practice

3 Large Systems

The Formalism

LS-BigDFT in Action

Fragments in BigDFT

4 In Practice

5 Summary

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

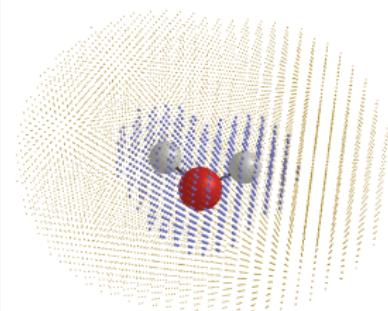
Code Inputs

Input Files

- **input parameters:** `name.yaml` (default is `input.yaml`), no specific order, full list of variables at https://gitlab.com/l_sim/bigdft-suite/-/blob/devel/bigdft/src/input_variables_definition.yaml?ref_type=heads
- **(atomic coordinates:** `name.xyz` (default is `posinp.xyz`) – if no units given then bohr are assumed, can also use other formats)
- **(pseudopotential:** `psppar.H`)

h2o.yaml

```
import: linear
dft:
  ixc: LDA # from ABINIT or LibXC
  hgrids: 0.35 # wavelet grid spacing
  rmult: [6, 8] # extent of grids
lin_basis_params:
  O:
    nbasis: 4 # number of SFs
    rloc: 7.0 # SF radii
  H:
    nbasis: 1 # number of SFs
    rloc: 6.0 # SF radii
```



[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

PyBigDFT

Complex Workflows with PyBigDFT

- larger systems often → more complex workflows – use PyBigDFT
- system generation, manipulation and visualisation – fragments are **intrinsic building blocks** (atoms → fragments → systems)
- YAML I/O format ↔ python dictionaries
- running calculations – both locally and **remote job submission**

Input File

```
from BigDFT import Inputfiles as I,  
InputActions as A  
inp = I.Inputfile()  
inp.set_hgrid(0.35)  
inp.set_xc("PBE")
```

Running Calculations

```
from BigDFT import Calculators as C  
code = C.SystemCalculator(omp=2,  
                         mpi_run="mpirun -n 16", skip=True)  
run = code.run(input=inp,  
               posinp="h2o.xyz", name="h2o_cubic")
```

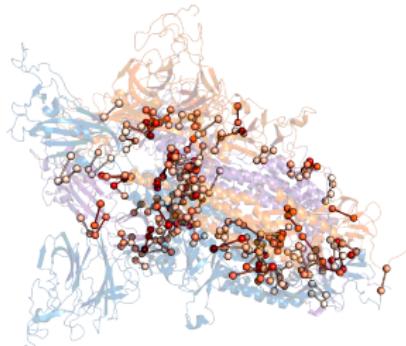
Post-Processing

```
from BigDFT import Logfiles as lf  
log = lf.Logfile("log-cubic.yaml")  
energy = log.energy  
time = log.log["Timings for root process"]["Elapsed time (s)"]
```

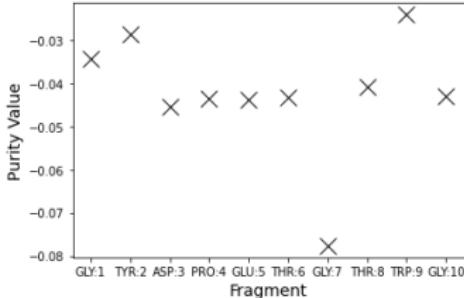
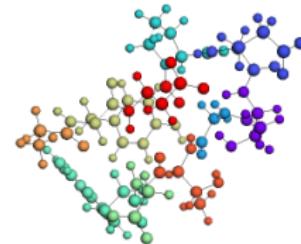
Example Workflow

Complexity Reduction

- PyBigDFT → routine analysis of complex biosystems
- combine with remote manager to submit datasets, e.g. different snapshots from MD
- interoperability with other packages e.g. ASE, OpenBabel, py3Dmol



```
from BigDFT.IO import read_pdb
psys = read_pdb(StringIO(istr))
psys.display()
plog = code.run(sys=psys, input=inp, name="luao", run_dir="work")
p = tool.run_compute_purity(psys, plog)
fig, axs = plt.subplots()
axs.plot(list(p.values()), 'kx', markersize=14)
axs.set_ylabel("Purity Value", fontsize=14)
axs.set_xticks(range(len(p)))
axs.set_xticklabels(list(p))
axs.set_xlabel("Fragment", fontsize=14)
plt.show()
```



What are Wavelets?

Haar Wavelets

Daubechies Wavelets

BigDFT

DFT with Daubechies Wavelets

BigDFT in Practice

Large Systems

The Formalism

LS-BigDFT in Action

Fragments in BigDFT

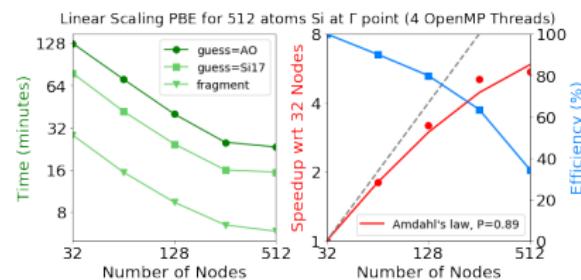
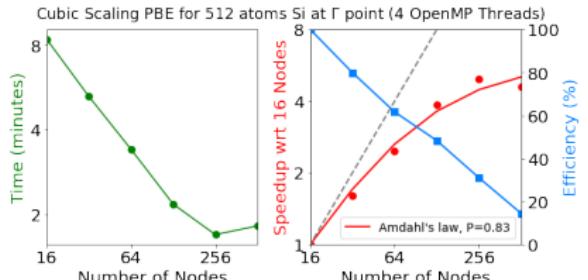
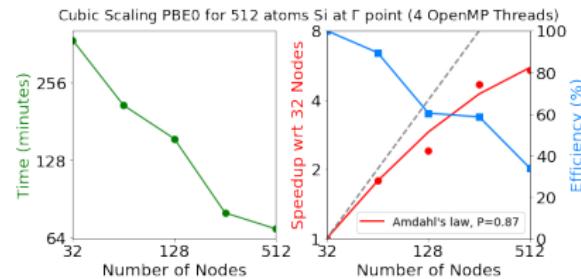
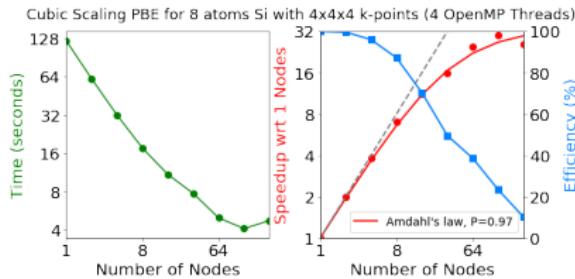
In Practice

Summary

Parallelisation

Bulk Si on Archer (24 cores/node)

- upper limit of 1 KS orbital/1 SF per MPI
- e.g. bulk Si – below crossover point (not a good system for LS-DFT)
- save time by generating SF guess from small cluster of Si, or use fragment approach



What are
Wavelets?

Haar Wavelets

Daubechies
Wavelets

BigDFT

DFT with
Daubechies
Wavelets

BigDFT in Practice

Large Systems

The Formalism

LS-BigDFT in Action

Fragments in
BigDFT

In Practice

Summary

Outline

1 What are Wavelets?

Haar Wavelets

Daubechies Wavelets

2 BigDFT

DFT with Daubechies Wavelets

BigDFT in Practice

3 Large Systems

The Formalism

LS-BigDFT in Action

Fragments in BigDFT

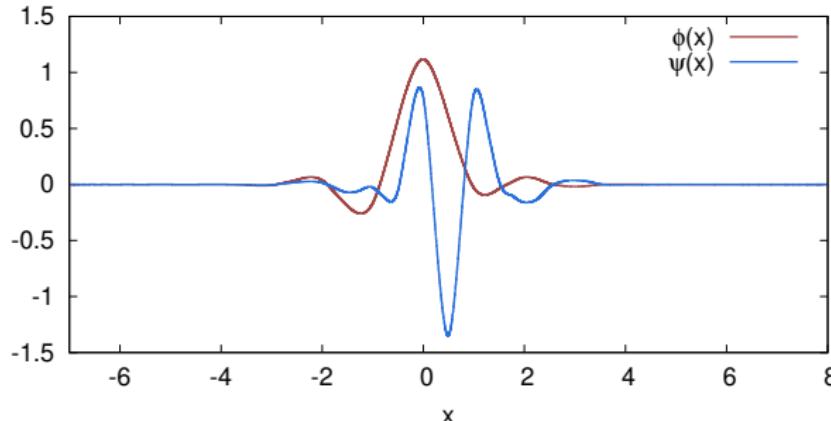
4 In Practice

5 Summary

BigDFT Summary

Daubechies wavelets → a DFT code which:

- exhibits systematic convergence
- is suitable for inhomogenous systems
- is flexible with respect to boundary conditions
- can treat very large systems
- efficiently exploits HPC
- has a range of functionalities (but not as comprehensive as e.g. CASTEP)



→ we retained many of the advantages of plane waves and Gaussians ✓

[What are Wavelets?](#)[Haar Wavelets](#)[Daubechies Wavelets](#)[BigDFT](#)[DFT with Daubechies Wavelets](#)[BigDFT in Practice](#)[Large Systems](#)[The Formalism](#)[LS-BigDFT in Action](#)[Fragments in BigDFT](#)[In Practice](#)[Summary](#)

Further Resources

- BigDFT website: <https://www.bigdft.org>
- BigDFT code: https://gitlab.com/l_sim/bigdft-suite
- Documentation:
https://l_sim.gitlab.io/bigdft-suite/index.html
- BigDFT school tutorials:
<https://github.com/BigDFT-group/bigdft-school>
- Videos from previous tutorials (CCPBioSim and ENCCS):
https://www.youtube.com/watch?v=f6P_guGl0zs
<https://www.youtube.com/watch?v=Ffa6QrMWi9g>
- Main reference paper: Ratcliff, Dawson, Fisicaro, Caliste, Mohr, Degomme, Videau, Cristiglio, Stella, D'Alessandro, Goedecker, Nakajima, Deutsch & Genovese, *J. Chem. Phys.* **152**, 194110 (2020)