PARALLELISATION OF THE ONETEP* LINEAR-SCALING PLANE WAVE PSEUDOPOTENTIAL DENSITY FUNCTIONAL THEORY PROGRAM

Part 1: CALCULATION OF OPERATOR MATRIX ELEMENTS

Chris-Kriton Skylaris

UNIVERSITY OF CAMBRIDGE
CAVENDISH LABORATORY

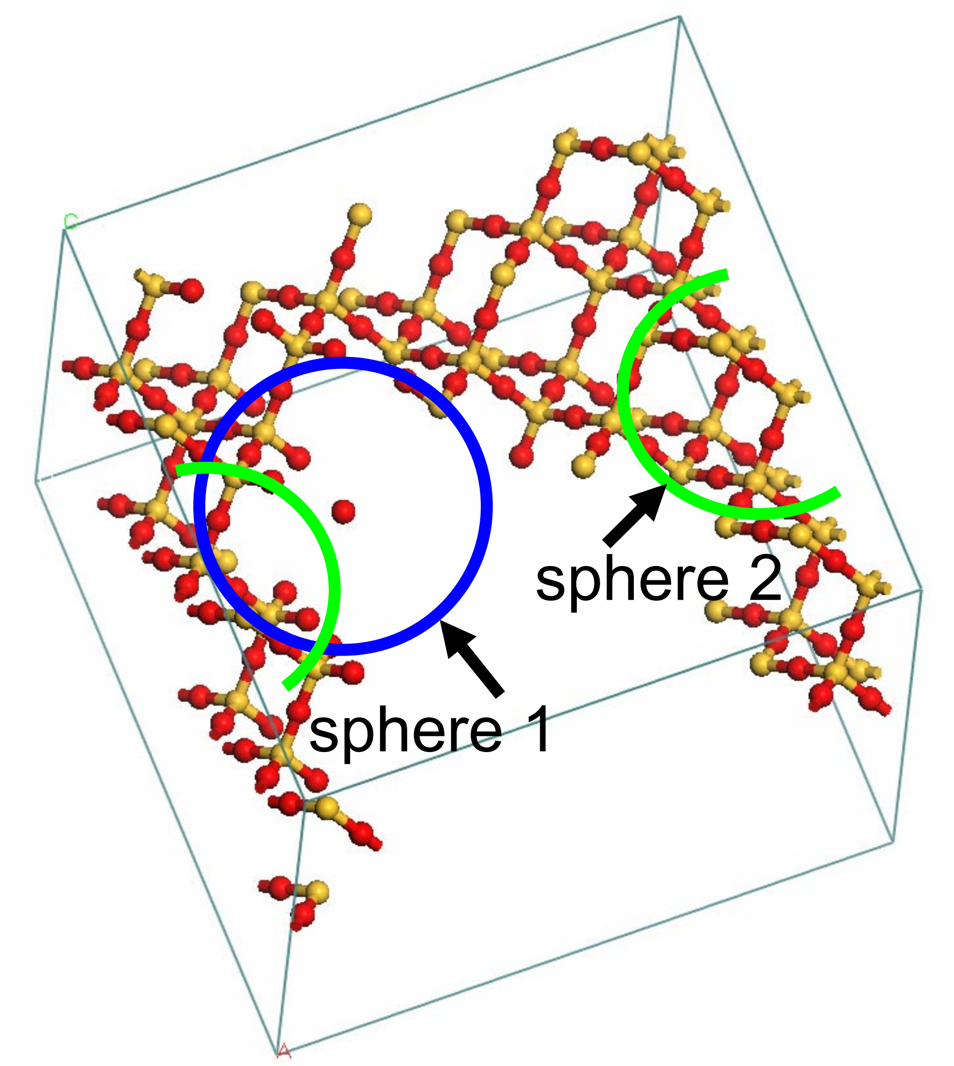
Acknowledgements

Peter D.Haynes

Arash A. Mostofi

Mike C. Payne

*ONES is now called ONETEP



Linear cost: Instead of optimising delocalised wavefunctions, optimise an equivalent set of Non-orthogonal Generalised Wannier Functions (NGWFs) which are strictly localised in real space.

Example:

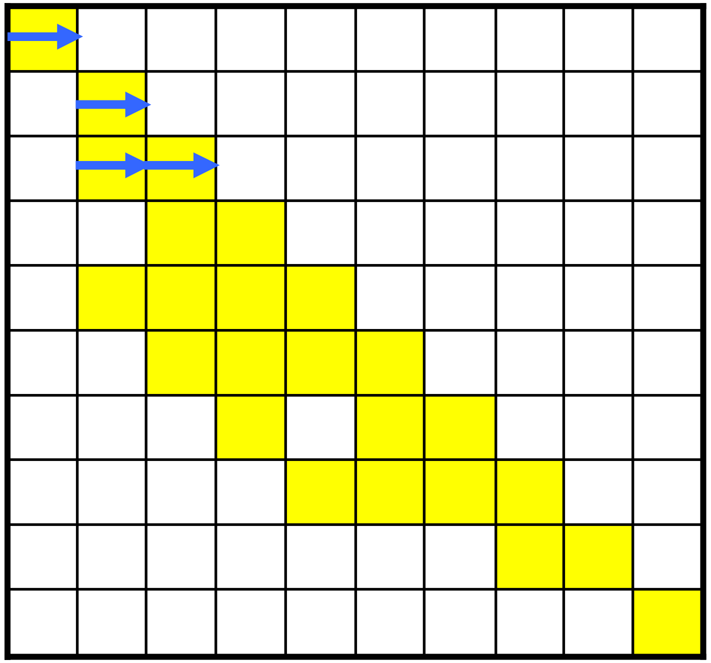
ϕ_α, ϕ_β, ϕ_γ, ϕ_δ are limited to sphere 1.

ϕ_ε, ϕ_ζ, ϕ_η are limited to sphere 2. Etc…

Need Hamiltonian matrix in NGWF representation

$<\varphi_\alpha|H|\varphi_\beta>$ is Sparse and Symmetric.

Serial code: Loop over rows and calculate only non-zero elements of lower triangle. Simple…

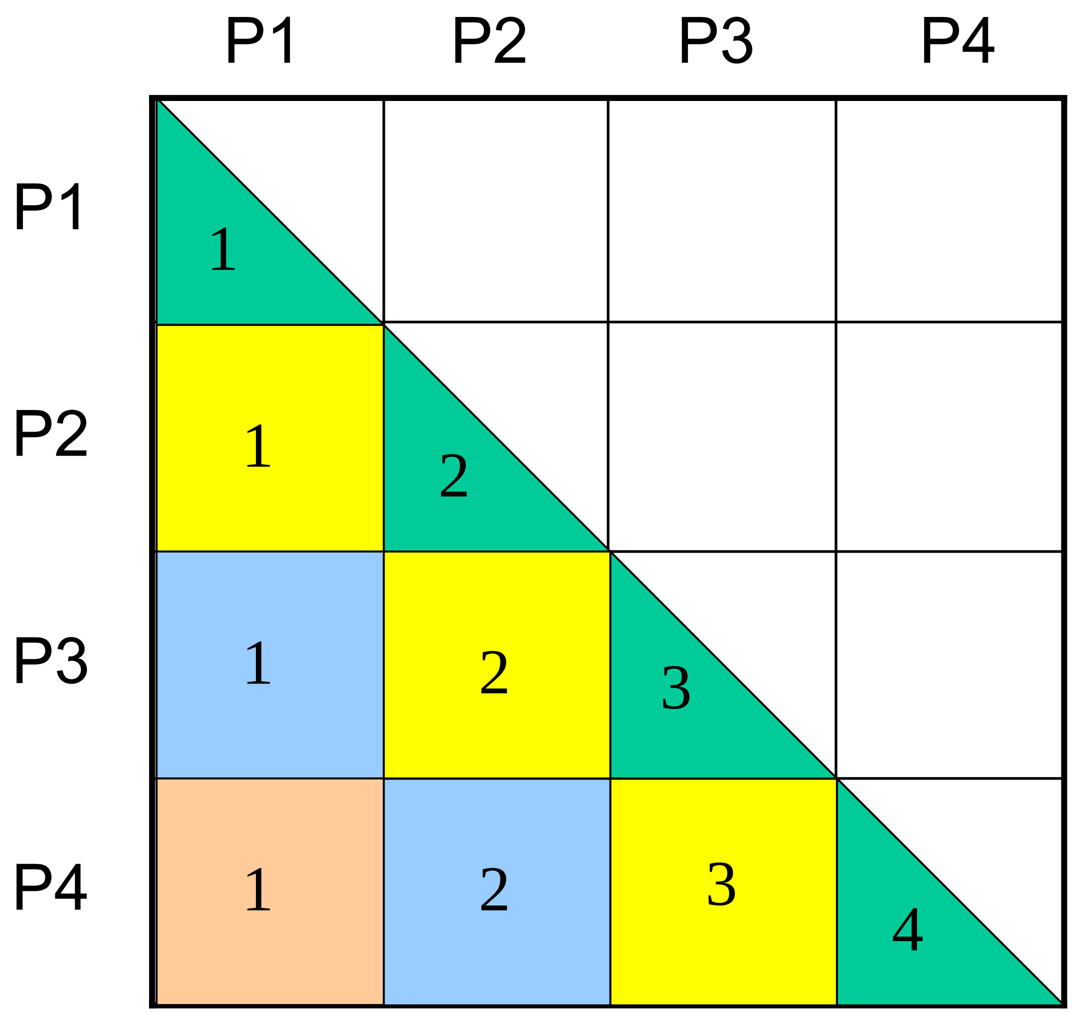


Parallel code: Not possible to store all NGWFs $\{\varphi\}_{All}$ on all processors.

Each processor stores a subset $\{\varphi\}_P$ of the NGWFs.

Matrix elements between NGWFs of different processors require communication.

$\langle\varphi_\alpha|H|\varphi_\beta\rangle$ example on 4 processors. Most expensive part is done locally on each processor: Calculate and store $|H\varphi\rangle$ for each $|\varphi\rangle$.



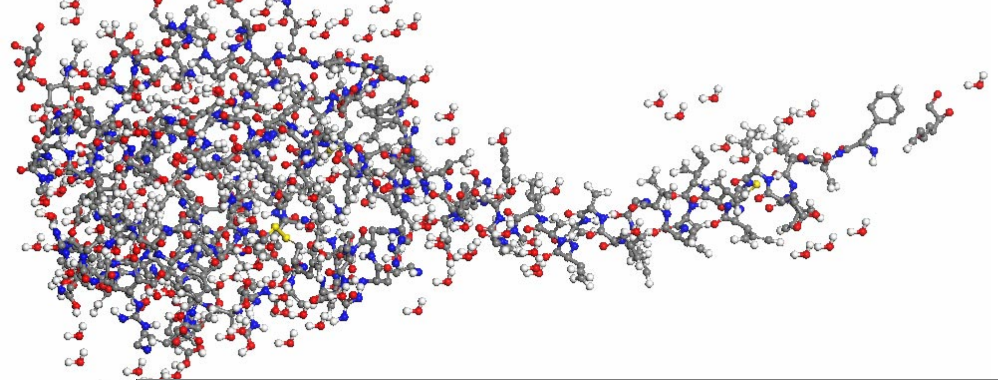
	P1	P2	P3	P4
P1	1			
P2	1	2		
P3	1	2	3	
P4	1	2	3	4

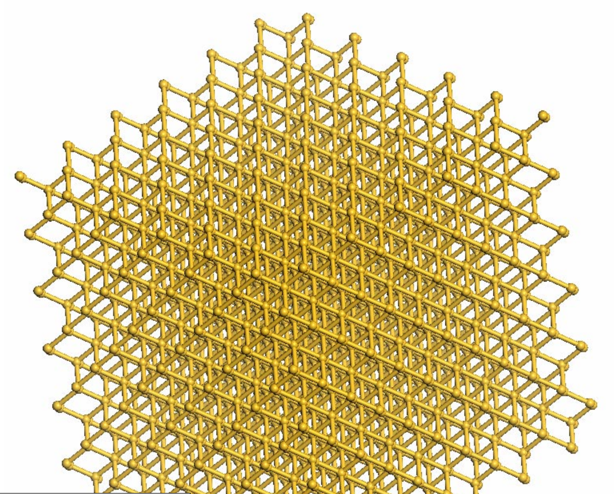
Calculate Px-Py sub-block of the matrix on processor y. For each row send $\langle\varphi|$ to only one other processor (y-n) and receive the relevant $\langle\varphi|$ from x.

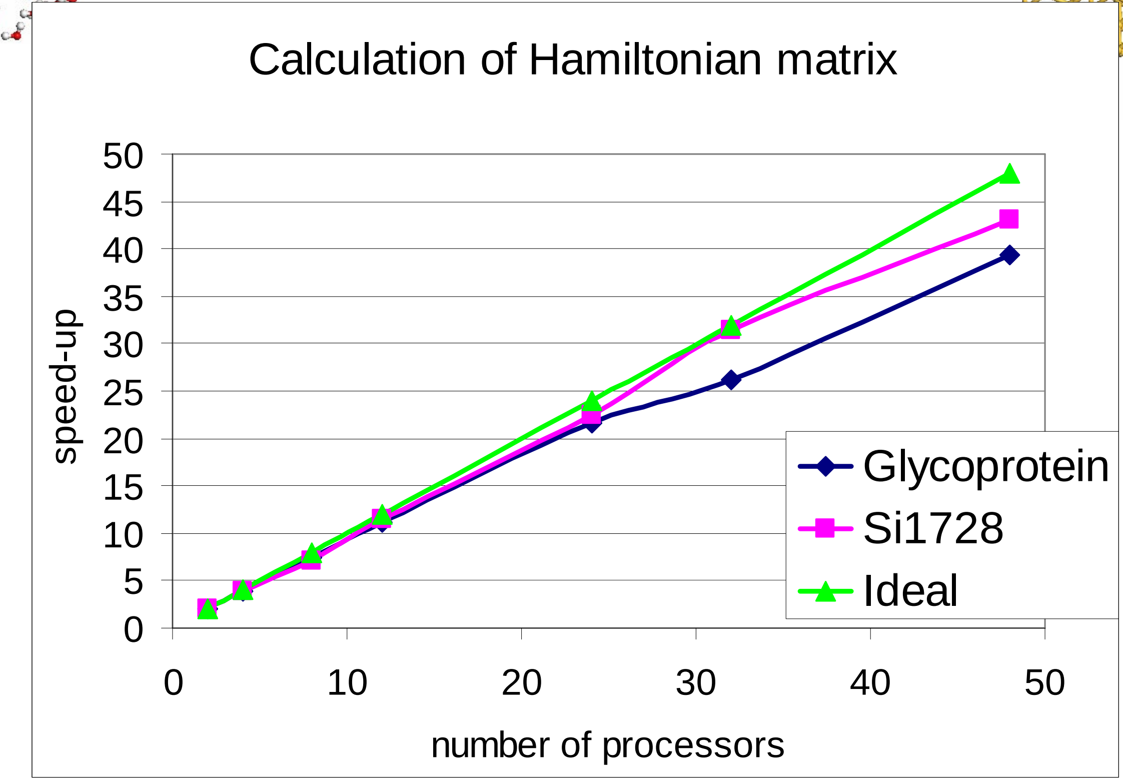
Send $\langle\varphi|$ in a “non-blocking” fashion so can complete local row calculation while send takes place.

Here steps n= 0, 1, 2, 3.

Works well!






Calculation of Hamiltonian matrix
speed-up
0
5
10
15
20
25
30
35
40
45
50
number of processors
0
10
20
30
40
50
◆ Glycoprotein
■ Si1728
▲ Ideal

WORK IN PROGRESS: PARALLEL CALCULATION OF CHARGE DENSITY

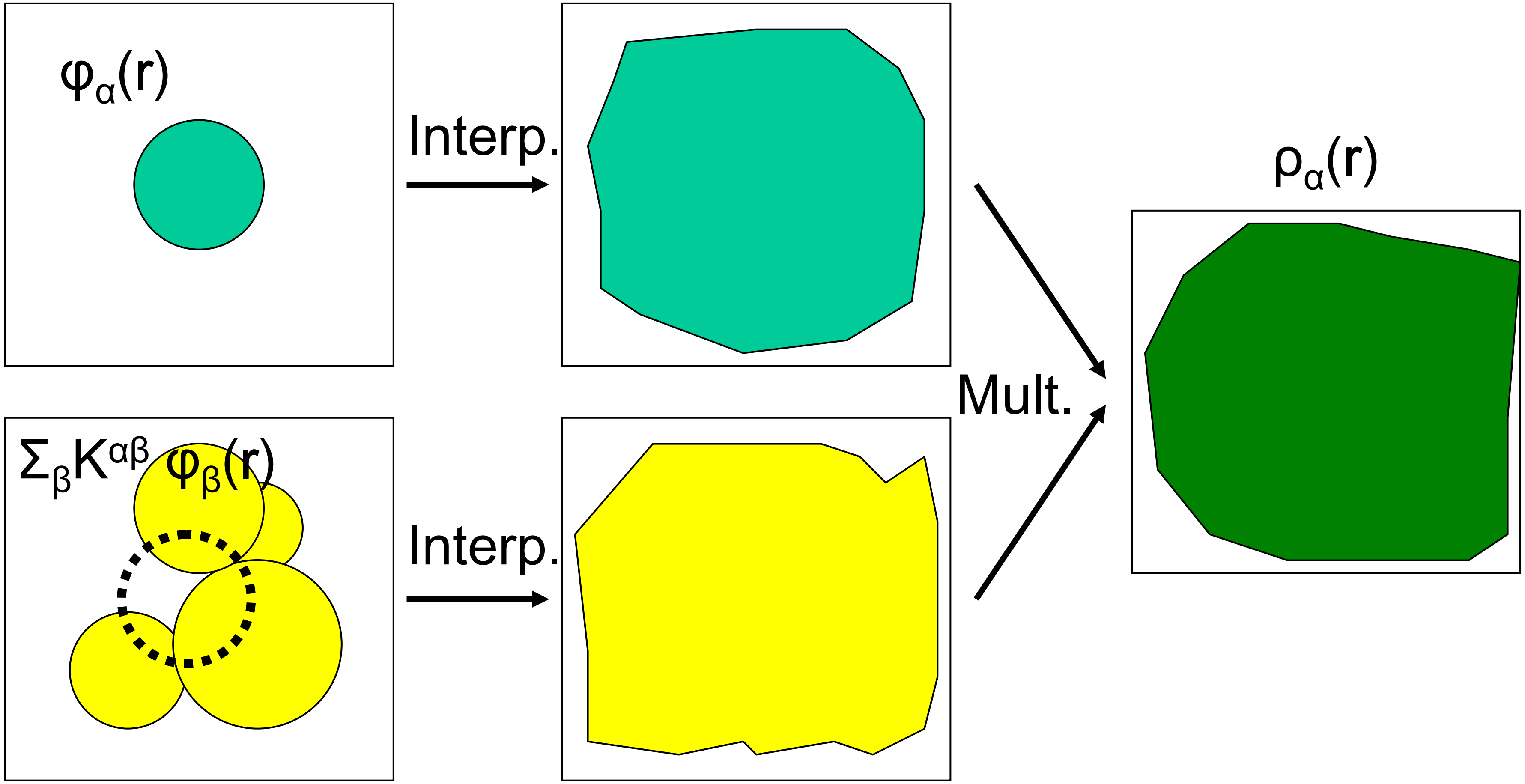
$$\rho(r)=\Sigma_{\alpha\beta}\ \varphi_\alpha(r)\ K^{\alpha\beta}\ \varphi_\beta(r)$$

Most expensive calculation part (Fourier interpolation of $\varphi_\alpha(r)$ and $\varphi_\beta(r)$ NGWF) takes place during (rather than before) communication.

The actual computation time still scales linearly with the number of processors but if we use the previous communication model the communication time scaling is unsatisfactory. Often processors are idle while waiting to receive a $\varphi_\alpha(r)$ from another processor. Need a different approach.

Calculation of one “row” of charge density

$$\rho_\alpha(\mathbf{r})=\varphi_\alpha(\mathbf{r})\ \Sigma_\beta K^{\alpha\beta}\ \varphi_\beta(\mathbf{r})$$



$\varphi_\alpha(\mathbf{r})$
Interp.
$\Sigma_\beta K^{\alpha\beta}\ \varphi_\beta(\mathbf{r})$
Interp.
Mult.
$\rho_\alpha(\mathbf{r})$