Band re-parallelization of DFPT in Abinit

Matthieu Verstraete
University of Liège Belgium

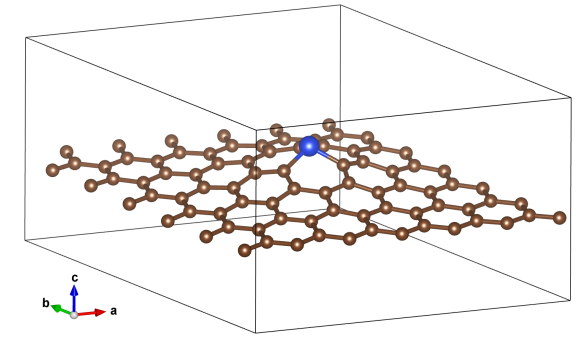
Abidev2021 online

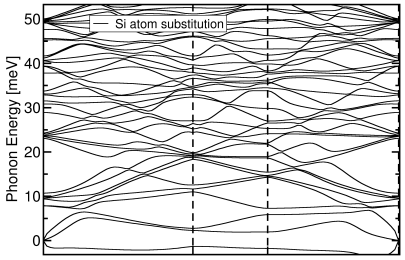
3rd of June 2021

Large scale DFPT

Problems with DFPT on big systems

- 3 natom calculations
- CPU time / scaling
- **memory**





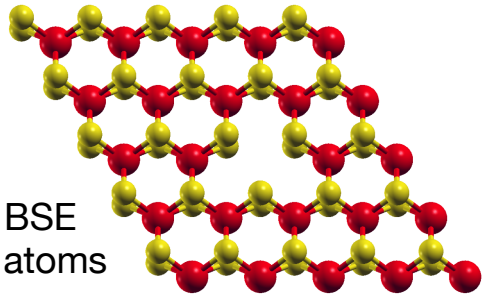
phonons
for 72 atoms

abinit

Tripathi *Nanoletters* (2018)

Similar issues in MBPT etc.

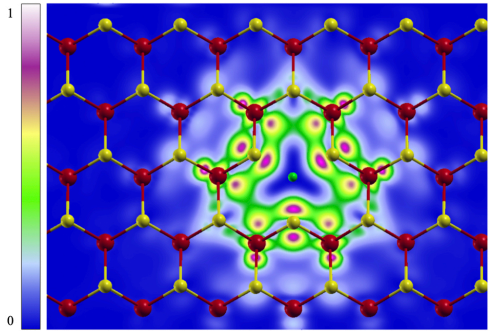
We want to mix the two!



GW BSE
on 74 atoms

QUANTUMESPRESSO

Yambo



Melo *Adv Quant Tech* (2021)

DFPT basics

Variational formulation of DFPT with Sternheimer equation

$$P_c \left(H^{(0)} - \varepsilon_i^{(0)} \right) P_c \left| \psi_i^{\alpha} \right\rangle = - P_c H^{\alpha} \left| \psi_i^{(0)} \right\rangle \qquad P_c = \sum_{j \in I^{\perp}} \left| \psi_j^{(0)} \right\rangle \left\langle \psi_j^{(0)} \right| = 1 - \sum_{j \in occ} \left| \psi_j^{(0)} \right\rangle \left\langle \psi_j^{(0)} \right|$$

Projector operation to impose gauge on ψ^{α}

Actually simpler than GS KS for orthonormalization: $\left\langle \psi_j^{(0)} \mid \psi_i^{\alpha} \right\rangle = 0 \quad \forall j$

Self consistency limited to $n^{(1)}$ / $H^{(1)}$ and $H^{(0)}$ is constant

Other non stationary expressions:

$$E^{\alpha\beta}_{\mathrm{el},-\mathbf{q},\mathbf{q}} \left\{ \psi^{(0)}; \psi^{\alpha}_{\mathbf{q}} \right\} = \frac{\Omega_0}{(2\pi)^3} \int_{\mathrm{BZ}} \sum_{m}^{\mathrm{occ}} s \left(\left\langle \psi^{\alpha}_{m\mathbf{k},\mathbf{q}} \left| v^{\beta}_{\mathrm{sep},\mathbf{k}+\mathbf{q},\mathbf{k}} \right| \psi^{(0)}_{m\mathbf{k}} \right\rangle + \left\langle \psi^{(0)}_{m\mathbf{k}} \left| v^{\alpha\beta}_{\mathrm{sep},\mathbf{k},\mathbf{k}} \right| \psi^{(0)}_{m\mathbf{k}} \right\rangle \right) d\mathbf{k} \quad + \frac{1}{2} \int_{\Omega_0} \left\{ \left[\bar{n}^{\alpha}_{\mathbf{q}}(\mathbf{r}) \right]^{*} \left[\bar{v}^{\beta}_{\mathrm{loc},\mathbf{q}}(\mathbf{r}) + \bar{v}^{\beta}_{\mathbf{xc0},\mathbf{q}}(\mathbf{r}) \right] \right\} d\mathbf{r}$$

$$+ \int_{\Omega_0} \left(n^{(0)}(\mathbf{r}) v^{\alpha^{*}\beta}_{\mathrm{loc}}(\mathbf{r}) \right) d\mathbf{r} + \frac{1}{2} \frac{d^2 E_{\mathrm{xc}}}{d\alpha_{-\mathbf{q}} d\beta_{\mathbf{q}}} \Bigg|_{n^{(0)}}$$

Gonze²+Lee PhysRevB **55** 10337, 10355 (1997)

Parallelization: the problem

Trivially k parallel (NB also $\psi^{(0)}_{k+q}$)

Band parallel for i, but not for j

FFT grid ~ parallel: lots of dot products

$$P_{ck}\left(H^{(0)}_k - \varepsilon^{(0)}_{ik}\right)P_{ck}\left|\psi^{\alpha}_{ik}\right\rangle = -P_{ck}H^{\alpha}_k\left|\psi^{(0)}_{ik}\right\rangle$$

$$P_{ck} = \sum_{j\in I^{\perp}k}\left|\psi^{(0)}_{jk}\right\rangle\left\langle\psi^{(0)}_{jk}\right| = 1 - \sum_{j\in occ_k}\left|\psi^{(0)}_{jk}\right\rangle\left\langle\psi^{(0)}_{jk}\right|$$

Load parallelization over i band index implemented (since forever)

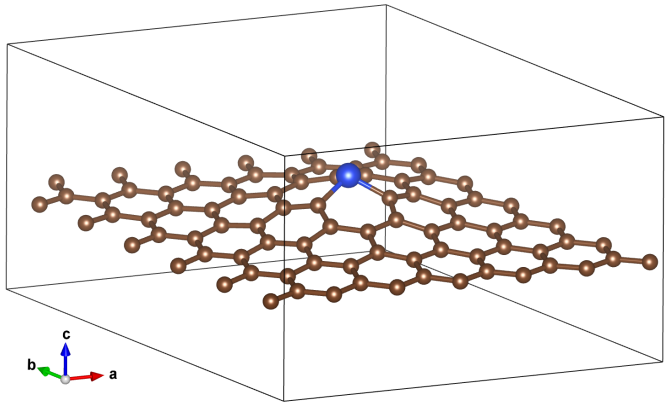
But **all GS bands are kept on each processor** : cg cgq cg1

So memory explodes as: 3 * nspinor * **mband** * **mpw** * **mkmem** * nsppol

Parallelization: possible solutions

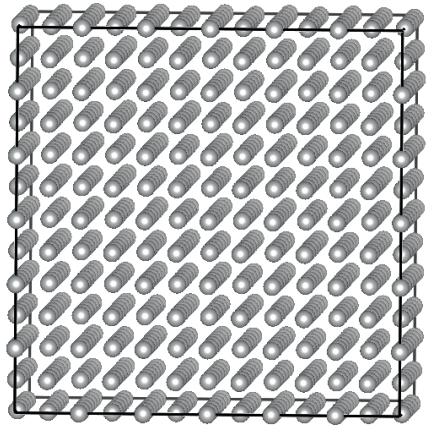
Memory explodes due to

- large cells (**mpw**)
- many atoms (**mband**)
- or both



What do we do?

1. openmp and just forget about it (limited by node RAM)
2. distribute FFT grid: complex + limited scaling
3. distribute band memory: I thought it was simpler…
4. or both ~ paral KGB (w/ transposition from band to FFT)



Band parallelization details

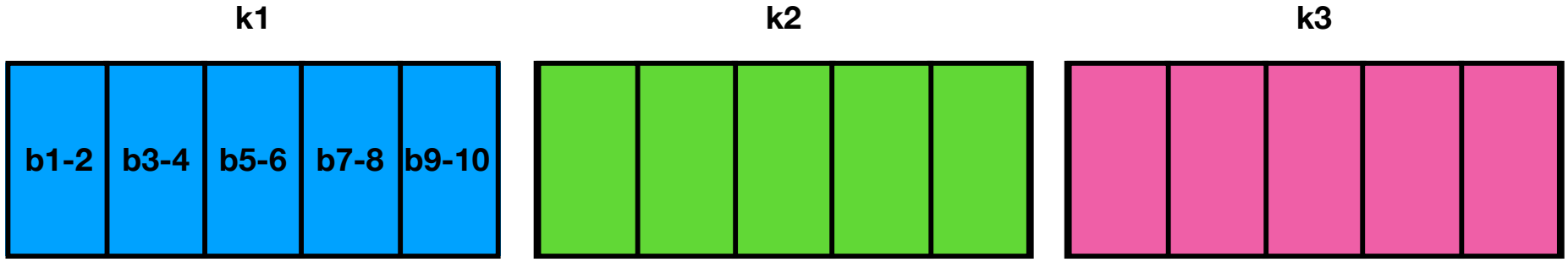
Limit bands per MPI thread for each cg cgq cg1: **mband_mem**

Routines modified: dfpt_ + scfcv → vtorho → vtowfk → cgwf

+ Non stationary expressions in dfpt_nstpaw nstwf…

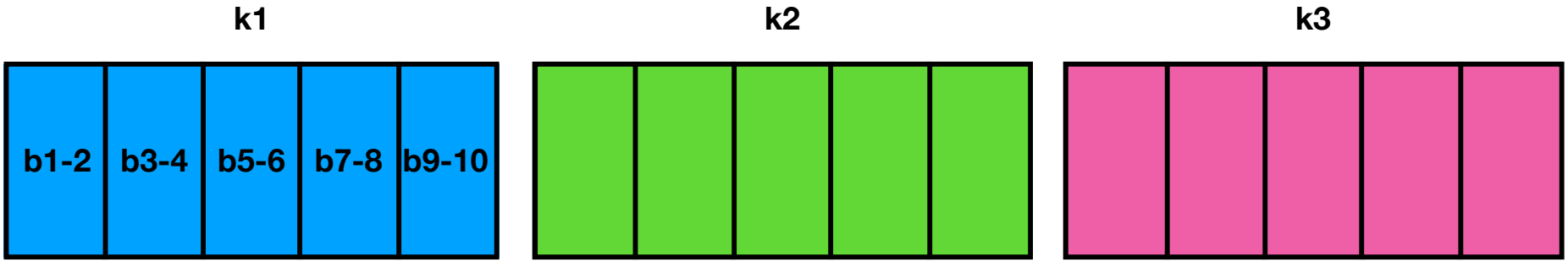
nproc/nkpt must give rectangular distribution (subcomm of kpt)

- !!! nkpt varies with each perturbation and spgroup !!!
- !!! tolerant in freezing out cpus which will not be used !!!

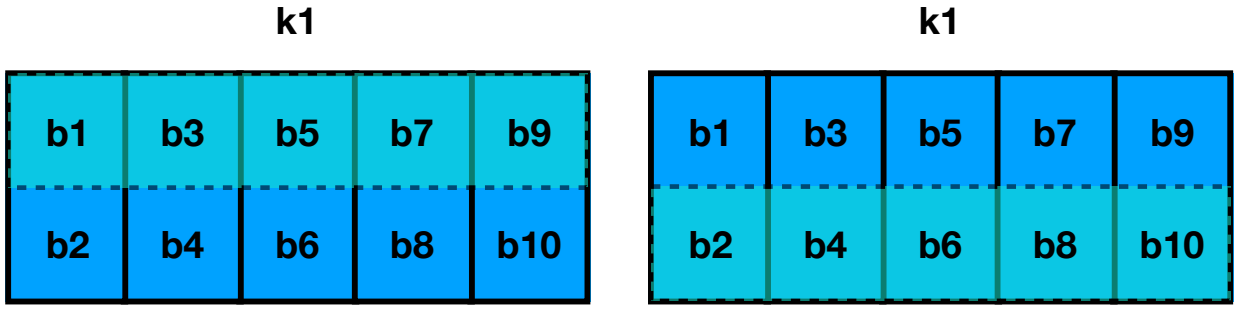


15 cores / 3 kpt = 5 which is divisor of 10 bands

Processor distribution



$$P_{ck} = 1 - \sum_{j \in occ_k} \left| \psi_{jk}^{(0)} \right\rangle \left\langle \psi_{jk}^{(0)} \right| = \sum_{m=1}^{mband_{mem}} \left[1 - \sum_{j \in group(m)} \left| \psi_{jk}^{(0)} \right\rangle \left\langle \psi_{jk}^{(0)} \right| \right] - (mband_{mem} - 1)$$



for i in group(m) broadcast $\psi_i^{(1)}$

$$|\bar{\psi}\rangle = \sum_{m=1}^{\text{mband}_{mem}} \left[\left(1 - \sum_{j \in group(m)} \left| \psi_{jk}^{(0)} \right\rangle \left\langle \psi_{jk}^{(0)} \right| \right) \right] |\psi_i^{(1)}\rangle$$

allreduce $\bar{\psi}$

$$\psi_i^{(1)} = \bar{\psi} - [\text{mband}_{mem} - 1]\psi_i^{(1)}$$

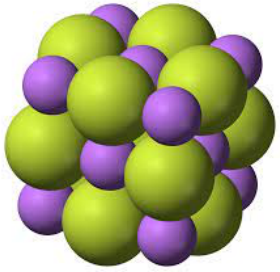
Profiling & performance

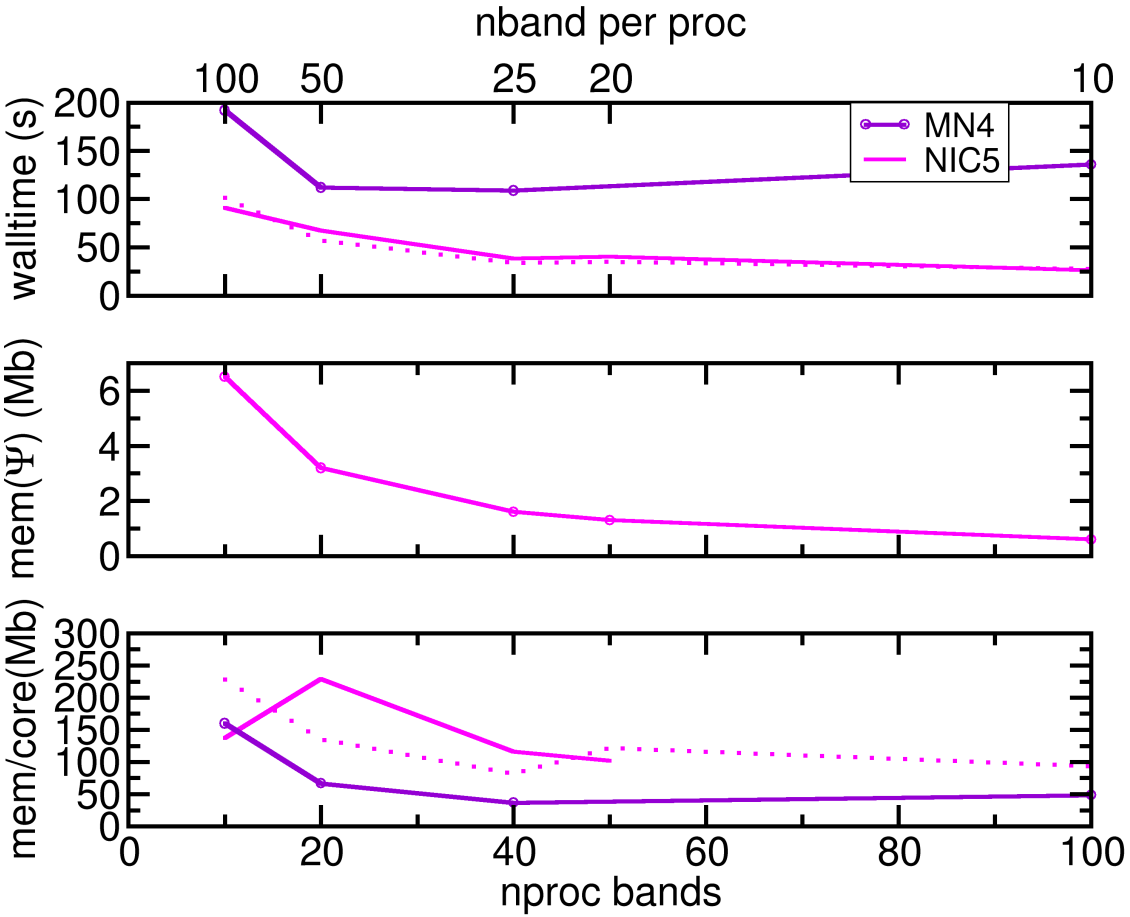
First profiling tests (thanks Joao!)

LiF 2 atoms 1 k; 1 pert; 1000 bands

Walltime still going down @ 100 cores:

- Efficiency limit ~ 10-20 bands/proc
- Depends on physical system
- + hardware & software





NB : we are trading communications + operations for memory!

Running now

Benchmarking, big cells, full memory profiling (partly done)

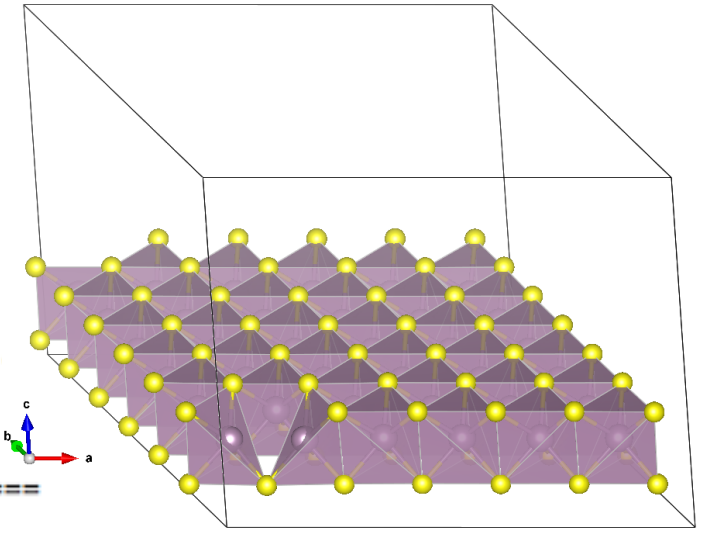
```
ITER STEP NUMBER    45
ETOT 45   755.09643976857      8.004E-10 6.203E-01 1.507E-10

At SCF step   45       vres2   =  1.51E-10 < tolvrs=  1.00E-09 =>converged
```

```
================================================================================
P This job should need less than                36362.033 Mbytes of memory.
  Rough estimation (10% accuracy) of disk space for files :
_ WF disk file :  -1857.912 Mbytes ; DEN or POT disk file :     64.074 Mbytes.
================================================================================
```

```
top - 23:21:23 up 26 days,  2:26,  1 user,  load average: 12.04, 12.01, 12.00
Tasks: 743 total,  13 running, 730 sleeping,   0 stopped,   0 zombie
%Cpu(s): 25.0 us,  0.0 sy,  0.0 ni, 75.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:  98621920 total, 34427080 used, 64194844 free,     1080 buffers
KiB Swap:  3905532 total,   154364 used,  3751168 free.   409952 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
377207 prleme01  20   0 2996128 2.360g  33072 R 100.33 2.509   1186:40 abinit
```



MoS2 with **SOC**
107 atoms, 960 bands, 4 k
ecut 40

768 x 20h = 15kcore hours
768 = 4 (k) x 192 (band)

TODO list

Action	Address	Size[b]	File	Line	Total Memory [bits]
mix%f_fftgr	A	8599633920	m_ab7_mixing.F90	554	
ph3d	A	5307487616	m_d2frnl.F90	618	11136646048
work	A	1084930560	m_getghc.F90	331	
wfraug1	A	1084930560	m_dfpt_mkrho.F90		

= 1GB

= 128 MB

Next frontiers:

- many v(nfft) and n(nfft) + other stuff present in the code
- npulayit instances → spread by proc, & do predictor/mixing steps in parallel there (usually trivially parallel in r)
- full paralKGB? Speedup limited to few (8?) FFT threads

  → Better openmp at this level (already in many places)
- 2d + vacuum & GGA