SELECTED INVERSION WITH APPLICATION IN ELECTRONIC STRUCTURE CALCULATIONS

Chao Yang

Computational Research Division
Lawrence Berkeley National Lab
Berkeley, CA, USA

Joint work with Lin Lin at LBNL
Funded by DOE

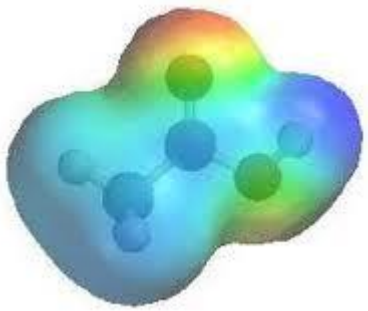
BERKELEY LAB
Lawrence Berkeley National Laboratory

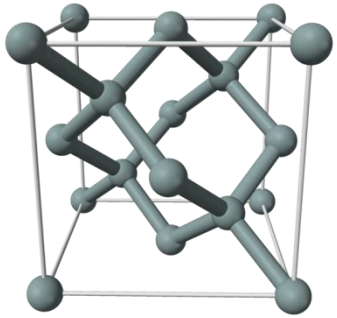
Outline

- Electronic structure and Kohn-Sham nonlinear eigenvalue problem
- Selected inversion
- Sequential implementation
- Parallel implementation

Electronic Structure of Matter and Modern Materials Design

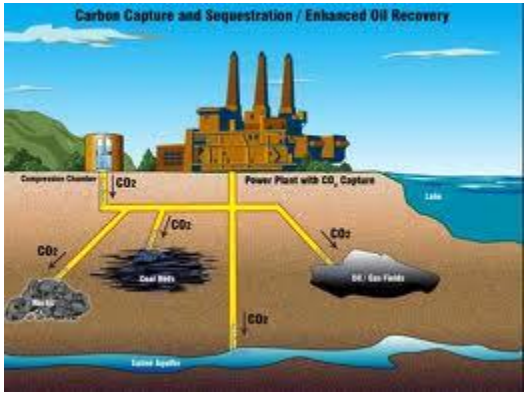
- Many-body Schrödinger's equation $H\Psi = \Psi E$





PV Solar Cells

Energy storage



Carbon capture

Kohn-Sham DFT: A Single Quasi-Particle Picture

- Total energy minimization

$$\min E_{tot}\left[\{\psi_i\}_{i=1}^{N}\right] = \frac{1}{2}\sum_{i=1}^{N}\int dx\, |\nabla\psi_i(x)|^2 + \int dx\, V_{ion}(x)\rho(x) + \frac{1}{2}\int dx \int dx' \frac{\rho(x)\rho(x')}{|x-x'|} + E_{xc}[\rho]$$

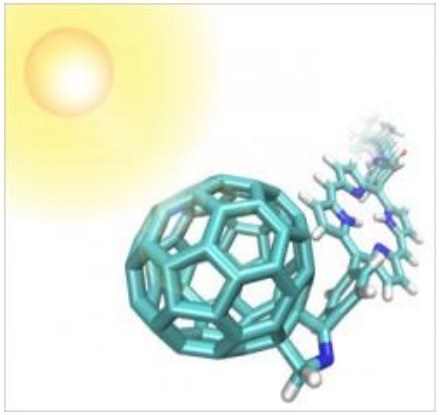
$$\rho(x) = \sum_{i=1}^{N} |\psi_i(x)|^2, \qquad \int dx\, \psi_i^*(x)\psi_j(x) = \delta_{ij}, \qquad x \in \mathbb{R}^3$$

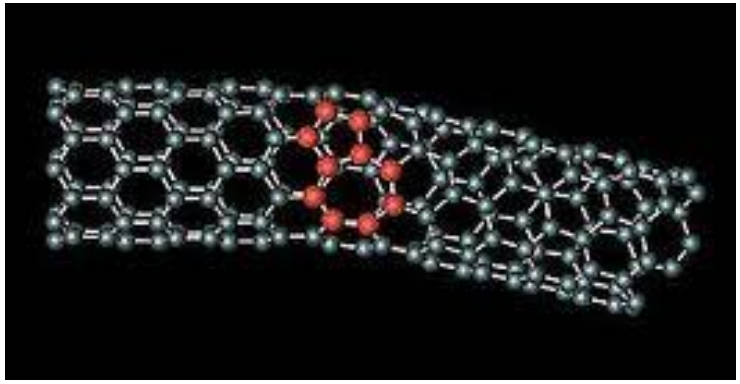
- Euler-Lagrange equation

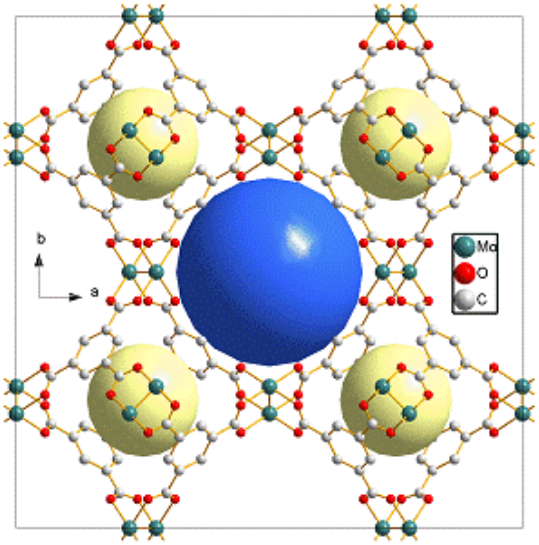
$$H[\rho]\psi_i(x) = \left(-\frac{1}{2}\Delta + V_{ion} + \int dx' \frac{\rho(x')}{|x-x'|} + V_{xc}[\rho]\right)\psi_i(x) = \varepsilon_i\psi_i(x)$$

The Challenge

- Nonlinearity and convergence
- Computational complexity



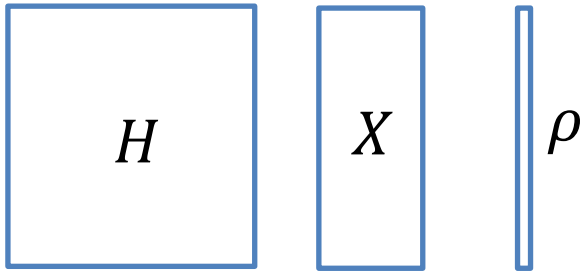


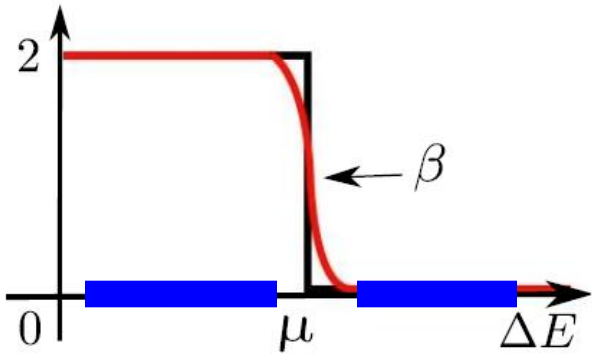


Charge density

$$H\big(\rho(X)\big)X = X\Lambda,\ X^TX = I,\ \rho(X) = \mathrm{diag}(XX^T)$$

where $H\big(\rho(X)\big) = L + V_{ion} + \mathrm{Diag}(L^{-1}\rho) + V_{xc}(\rho)$



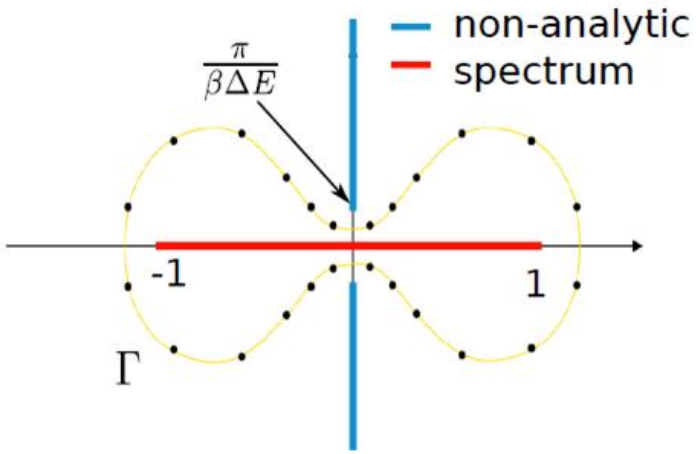


$$\rho = \begin{cases} \mathrm{diag}(XX^T) \\ \mathrm{diag}\left[(X, X^{\perp})\begin{pmatrix} I & \\ & 0 \end{pmatrix}\begin{pmatrix} X^T \\ X^{\perp T} \end{pmatrix}\right] \\ \mathrm{diag}\big(I + e^{\beta(H[\rho]-\mu)}\big)^{-1} \end{cases}$$

- $\beta = 1/k_BT$: inverse temperature
- μ: Chemical potential

Function Evaluation via Fermi Operator Expansion

- Pole expansion
  $\left(1 + e^{\beta(H-\mu I)}\right)^{-1} \approx Im\left[\sum_{i=1}^{n_p} \omega_i (H - z_i I)^{-1}\right]$



- Selected Inversion:
  - Compute the diagonal of $(H - z_i I)^{-1}$ without computing the full inverse
  - Applicable if *H* is sparse (i.e. not applicable to planewave discretization)
  - Need to perform sparse factorization of $H - z_i I$
  - Multiple levels of parallelism

$O(N)$ for 1D, $O\left(N^{3/2}\right)$ for 2D, $O(N^2)$ for 3D

L. Lin et al. 2010
Hale, Higham,Trefethen 2008
Saad & Sidjie 2008

Selected Inversion

- Given an LDL^T factorization

$$H = \begin{pmatrix} 1 & \\ \ell & I \end{pmatrix} \begin{pmatrix} \alpha & \\ & S \end{pmatrix} \begin{pmatrix} 1 & \ell^T \\ & I \end{pmatrix}$$

- Full inverse

$$H^{-1} = \begin{pmatrix} \alpha^{-1} + \ell^T S^{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$$

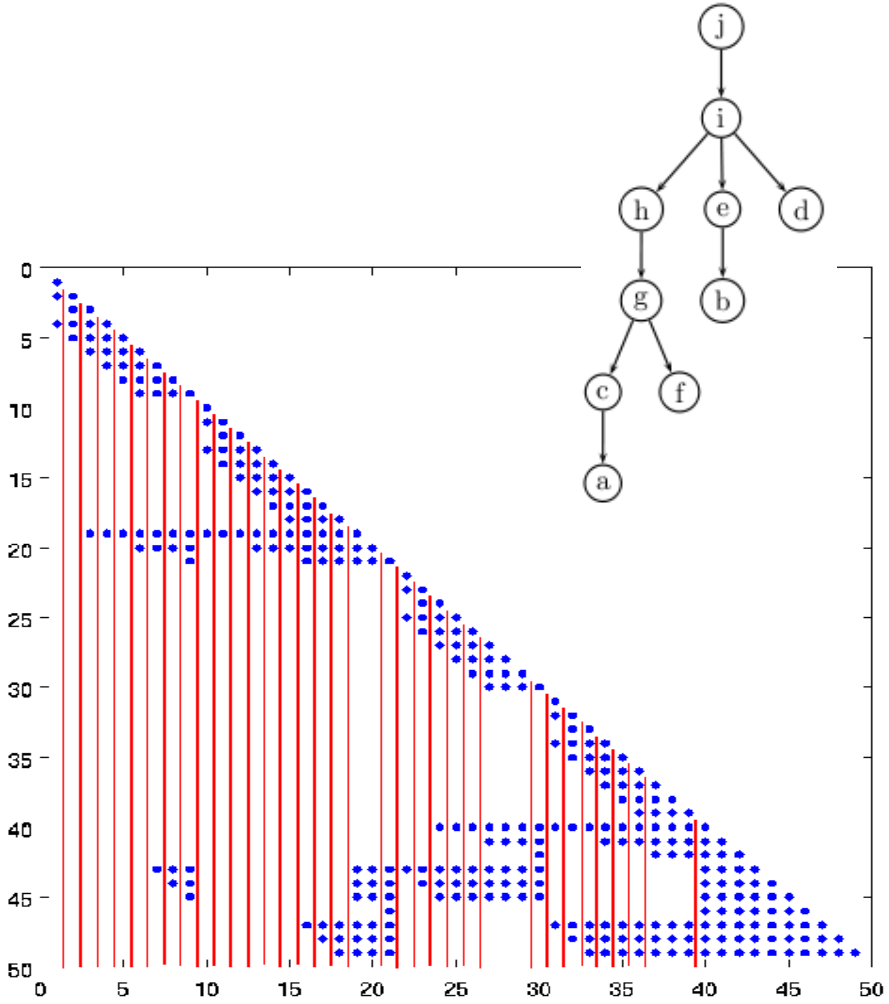
Observations:

- If ℓ is sparse, we do not need the entire S^{-1}in order to obtain the (1,1) entry of H^{-1};
- Complexity: $O(N)$ for 1D, $O\left(N^{3/2}\right)$ for 2D, $O(N^2)$ for 3D;

Takahashi 1973, Erisman & Tinney 1975,
Li et al (2008), Lin et al (2009), Amestoy et al, Ucar (2010)

Recursion, elimination tree, supernodes

- $H^{-1} = \begin{pmatrix} \alpha^{-1} + \ell^T S^{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$
- S^{-1} can be computed in a recursive fashion starting from the low right corner of the LDL^T factor
- In order to obtain the diagonal of H^{-1}, we only need to compute $S^{-1}_{i,j}$ for all (i,j) such that $L_{i,j} \neq 0$



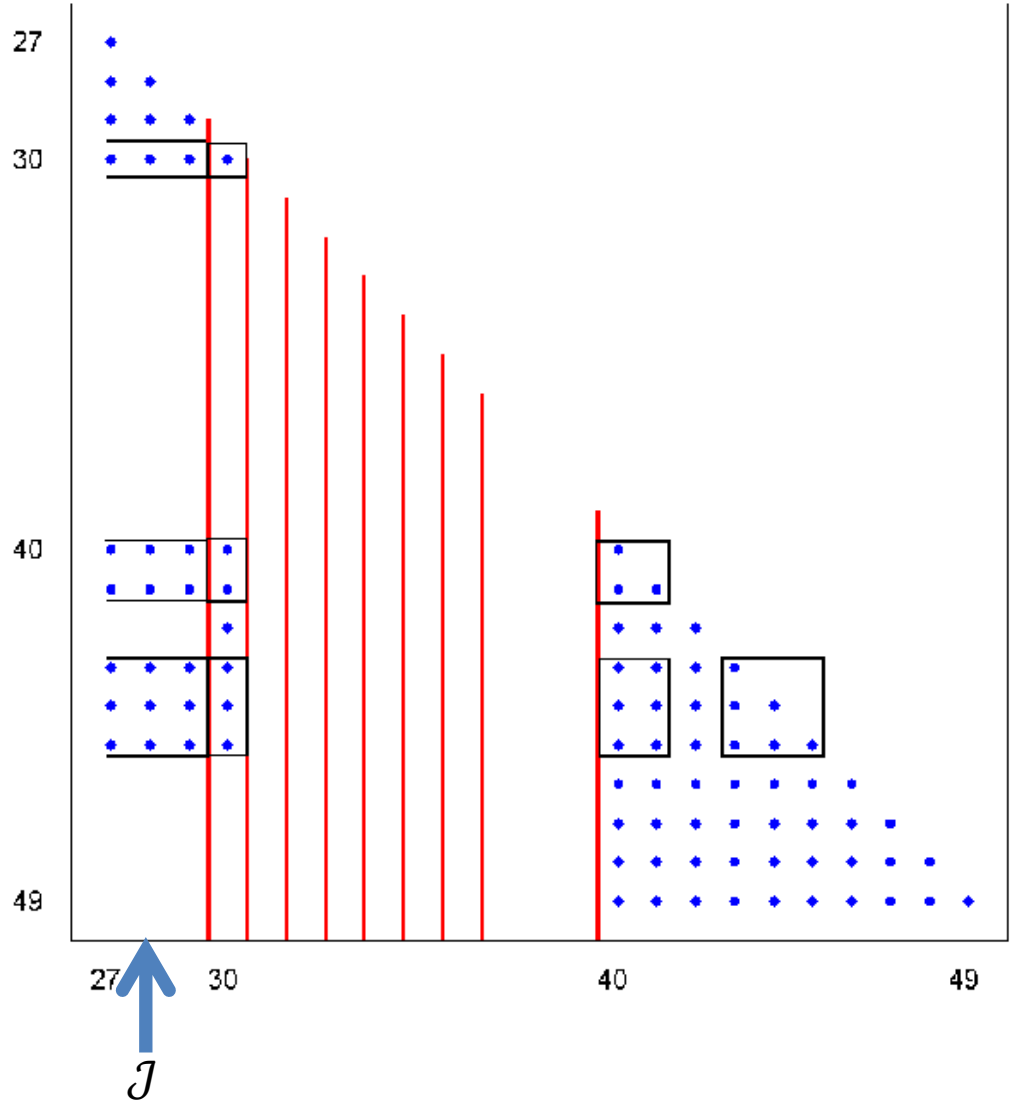
Right-looking Supernodal SelInv Algorithm

Input: (1) The supernode partition of columns of A: $\{1, 2, \ldots, n_{sup}\}$;
(2) A supernode LDL^T factorization of A;
Output: Selected elements of A^{-1}, i.e. $(A^{-1})_{i,j}$ such that $L_{i,j} \neq 0$.

1: Compute $A^{-1}_{n_{sup},n_{sup}} = D^{-1}_{n_{sup},n_{sup}}$;
2: **for** $\mathcal{J} = n_{sup} - 1, n_{sup} - 2, \ldots, 1$
3: Identify the nonzero rows in the $\mathcal{J}$-th supernode $S_{\mathcal{J}}$;
4: Perform $Y = A^{-1}_{S_{\mathcal{J}},S_{\mathcal{J}}} L_{S_{\mathcal{J}},\mathcal{J}}$;
5: Calculate $A^{-1}_{\mathcal{J},\mathcal{J}} = D^{-1}_{\mathcal{J},\mathcal{J}} + Y^T L_{S_{\mathcal{J}},\mathcal{J}}$;
6: Set $A^{-1}_{S_{\mathcal{J}},\mathcal{J}} \leftarrow -Y$;
7: **end for**

- $\mathcal{J}$: supernode index, set of indices belonging to the supernode
- $S_{\mathcal{J}}$: row indices of nonzero rows in the $\mathcal{J}$th supernode

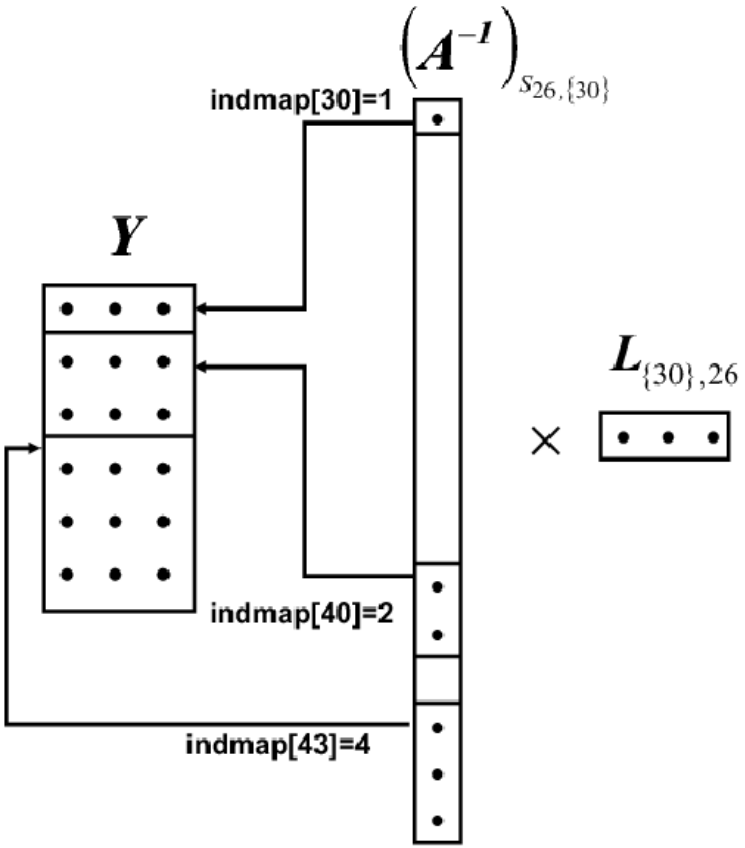
Example

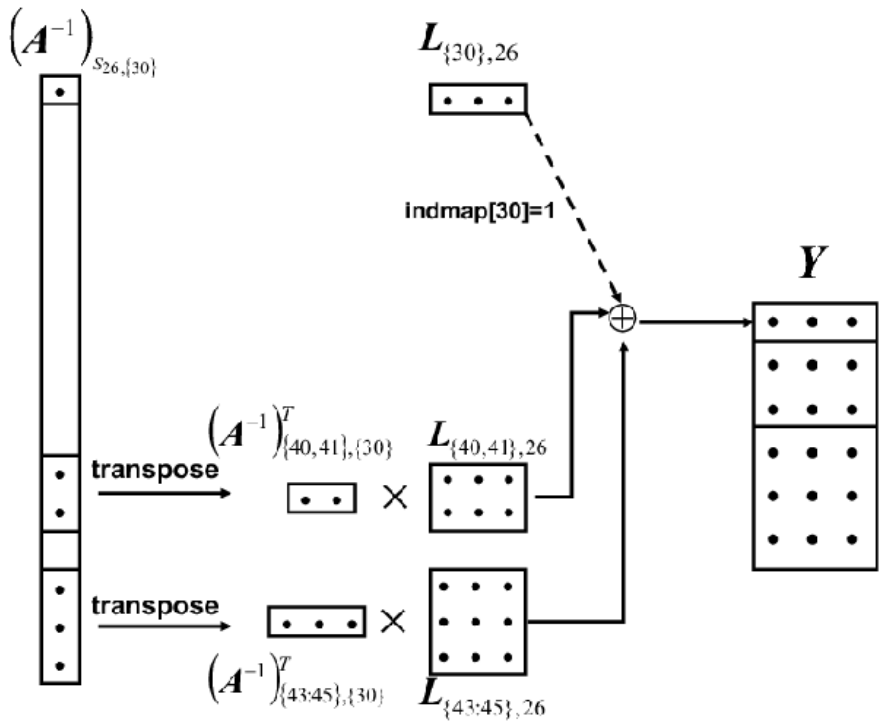


indmap[30]=1
indmap[40]=2
indmap[41]=3
indmap[43]=4
indmap[44]=5
indmap[45]=6

E. Ng & B. Peyton 1993

Fast multiplication





Algorithm

Input: (1) The $\mathcal{J}$-th supernode of L, $L_{S_\mathcal{J},\mathcal{J}}$, where $S_\mathcal{J}$ contains the indices of the nonzero rows in $\mathcal{J}$. The index set $S_\mathcal{J}$ is partitioned into disjoint $n_\mathcal{J}$ subsets of contiguous indices, i.e. $S_\mathcal{J} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{n_\mathcal{J}}\}$;
(2) The nonzero elements of A^{-1} that have been computed previously. These elements are stored in $L_{S_\mathcal{K},\mathcal{K}}$ for all $\mathcal{K} > \mathcal{J}$;

Output: $Y = (A^{-1})_{S_\mathcal{J},S_\mathcal{J}} L_{S_\mathcal{J},\mathcal{J}}$;

1: Construct an `indmap` array for nonzero rows in the $\mathcal{J}$-th supernode;
2: **for** $j = 1, 2, \ldots, n_\mathcal{J}$
3: Identify the supernode $\mathcal{K}$ that contains $\mathcal{I}_j$;
4: Let $\mathcal{R}_1 = \texttt{indmap}(\mathcal{I}_j)$;
5: Calculate $Y_{\mathcal{R}_1,*} \leftarrow Y_{\mathcal{R}_1,*} + (A^{-1})_{\mathcal{I}_j,\mathcal{I}_j} L_{\mathcal{I}_j,\mathcal{J}}$;
6: **for** $i = j+1, j+2, \ldots n_\mathcal{J}$
7: Use `indmap` to find the first nonzero row in the $\mathcal{K}$-th supernode that belongs to $\mathcal{I}_i$ so that $(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j}$ can be located;
8: Let $\mathcal{R}_2 = \texttt{indmap}(\mathcal{I}_i)$;
9: Calculate $Y_{\mathcal{R}_2,*} \leftarrow Y_{\mathcal{R}_2,*} + (A^{-1})_{\mathcal{I}_i,\mathcal{I}_j} L_{\mathcal{I}_j,\mathcal{J}}$;
10: Calculate $Y_{\mathcal{R}_1,*} \leftarrow Y_{\mathcal{R}_1,*} + [(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j}]^T L_{\mathcal{I}_i,\mathcal{J}}$;
11: **end for**
12: **end for**
13: Reset the nonzero entries of `indmap` to zero;

SelInv & Test Problems

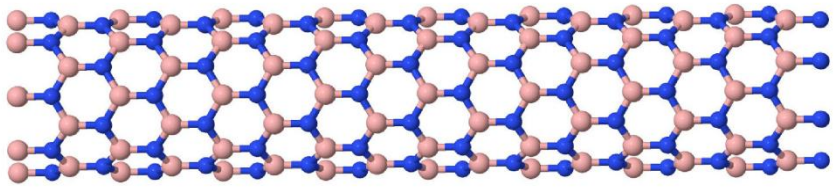
Problem	n	$\|A\|$	$\|L\|$
bcsstk14	1,806	32,630	112,267
bcsstk24	3,562	81,736	278,922
bcsstk28	4,410	111,717	346,864
bcsstk18	11,948	80,519	662,725
bodyy6	19,366	77,057	670,812
crystm03	24,696	304,233	3,762,633
wathen120	36,441	301,101	2,624,133
thermal1	82,654	328,556	2,690,654
shipsec1	140,874	3,977,139	40,019,943
pwtk	217,918	5,926,171	56,409,307
parabolic_fem	525,825	2,100,225	34,923,113
tmt_sym	726,713	2,903,837	41,296,329
ecology2	999,999	2,997,995	38,516,672
G3_circuit	1,585,478	4,623,152	197,137,253

Performance

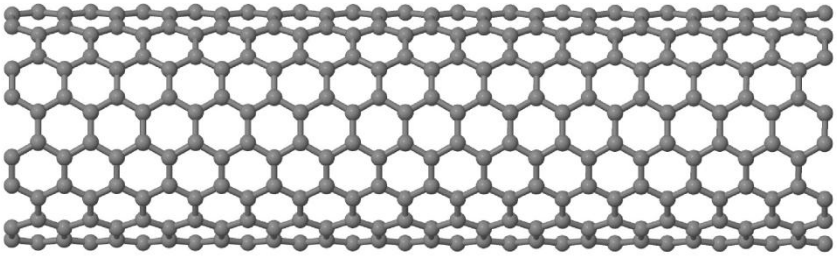
Problem	Selected inversion time (s)	Direct inversion time (s)	Speedup factor
bcsstk14	0.01 sec	0.13 sec	13
bcsstk24	0.02 sec	0.58 sec	29
bcsstk28	0.02 sec	0.88 sec	44
bcsstk18	0.24 sec	5.73 sec	24
bodyy6	0.09 sec	5.37 sec	60
crystm03	0.78 sec	26.89 sec	34
wathen120	0.34 sec	48.34 sec	142
thermal1	0.44 sec	95.06 sec	216
shipsec1	17.66 sec	3346 sec	192
pwtk	14.55 sec	5135 sec	353
parabolic_fem	20.06 sec	7054 sec	352
tmt_sym	13.98 sec	>3 hours	>772
ecology2	16.04 sec	>3 hours	>673
G3_circuit	218.7 sec	>3 hours	>49

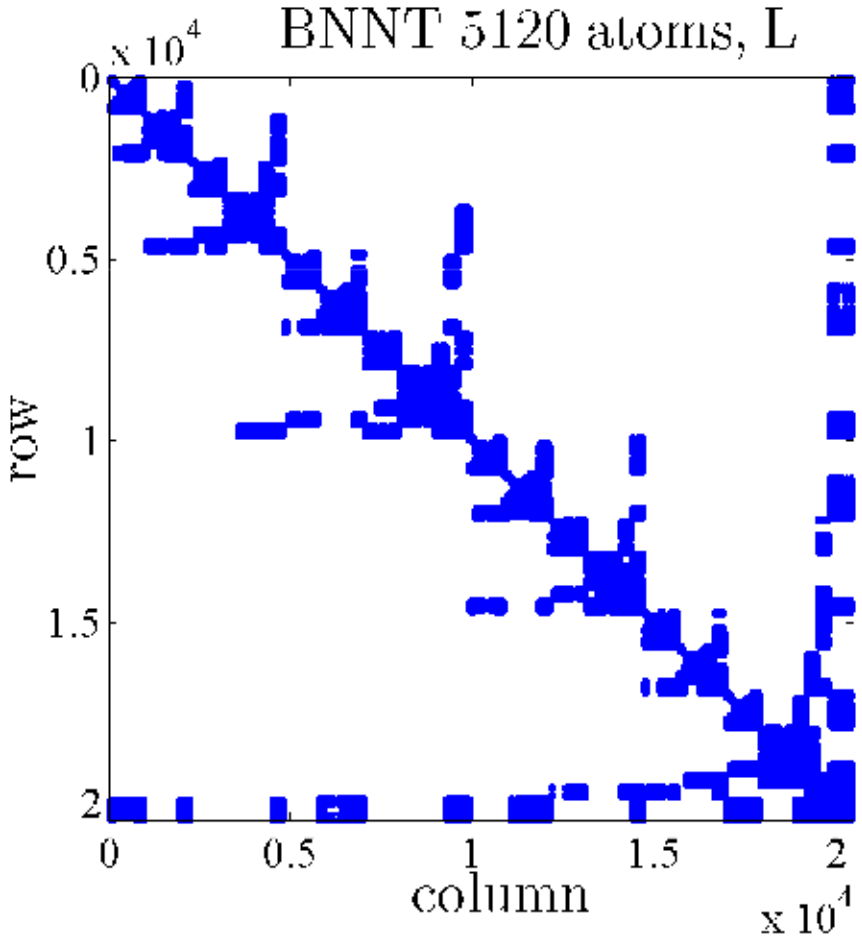
Nanotubes

Boron Nitride (insulator)

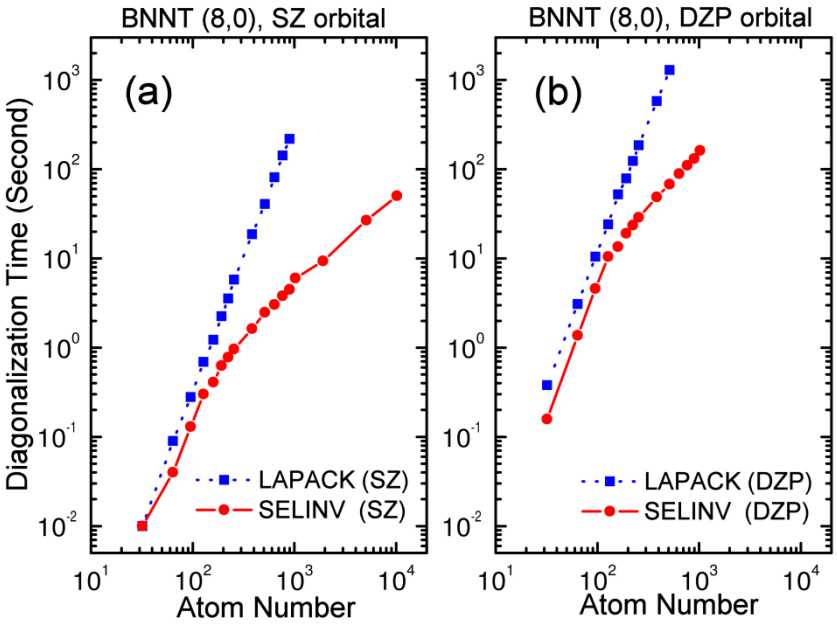


Carbon (metallic)

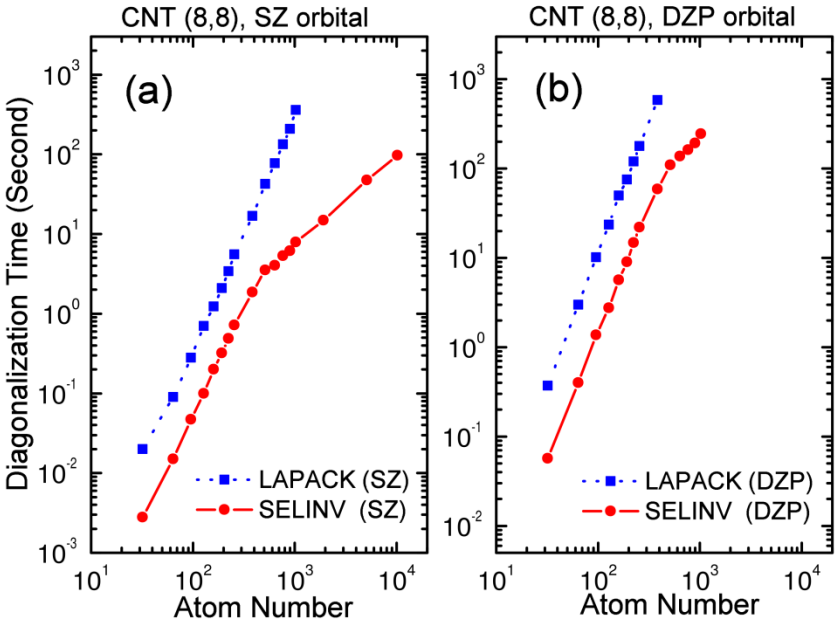




O(N) scaling

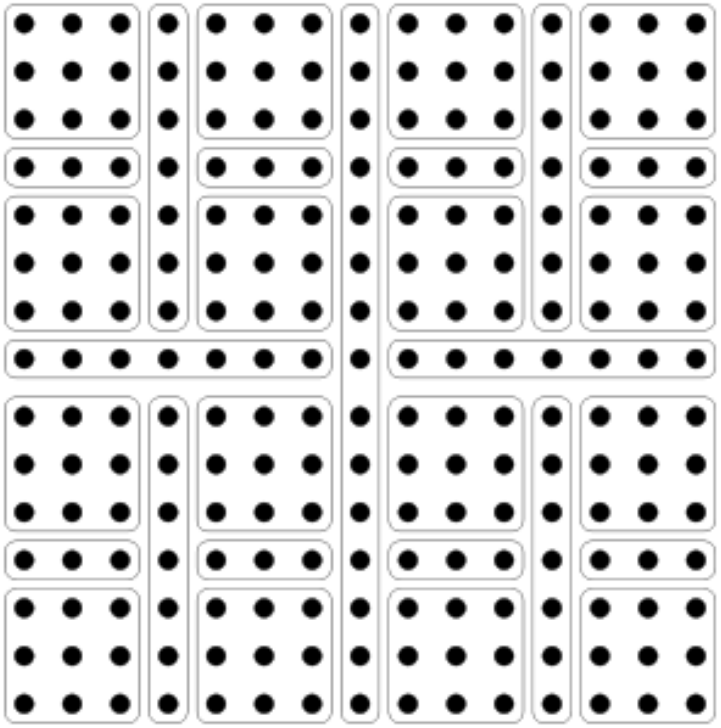


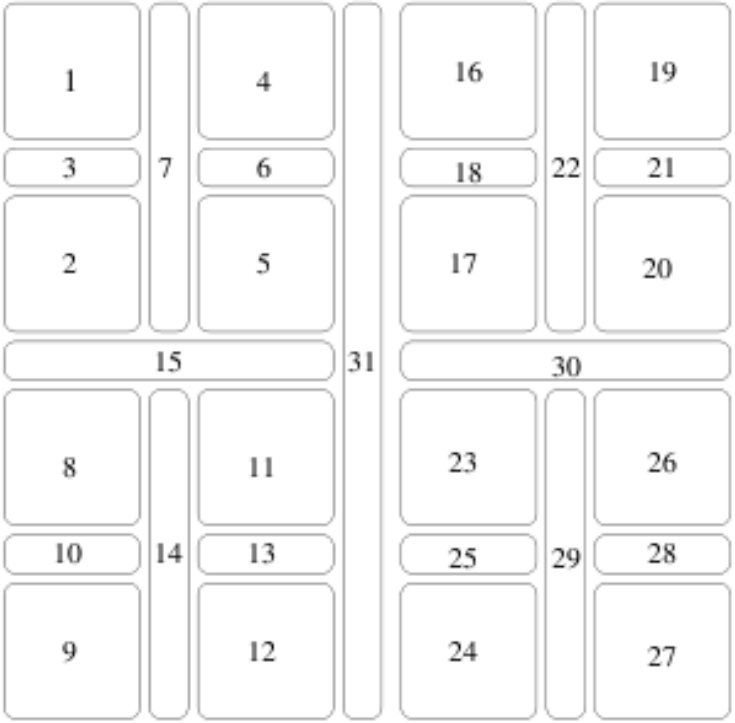
Boron nitride



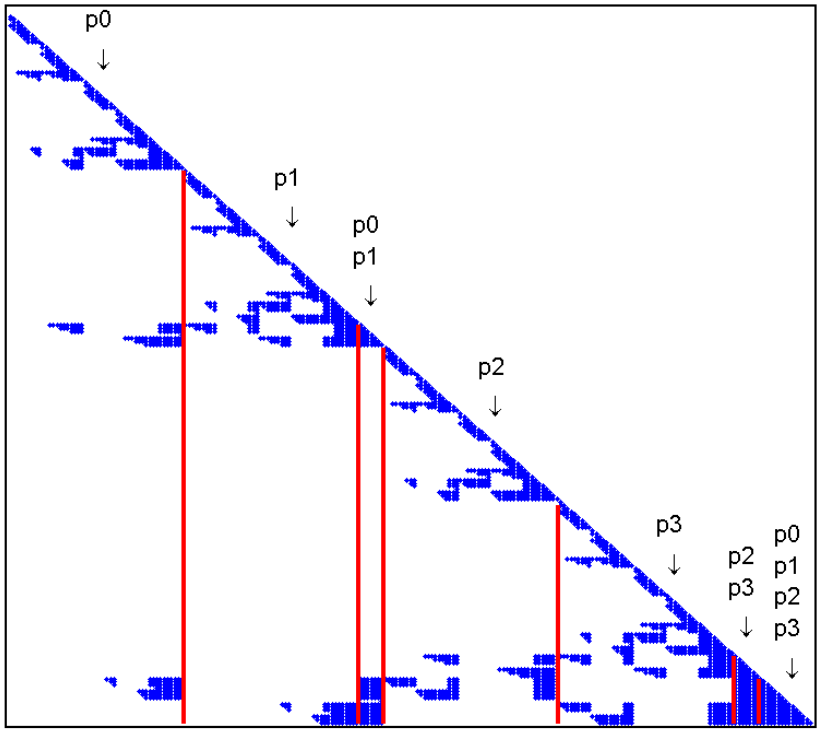
Carbon

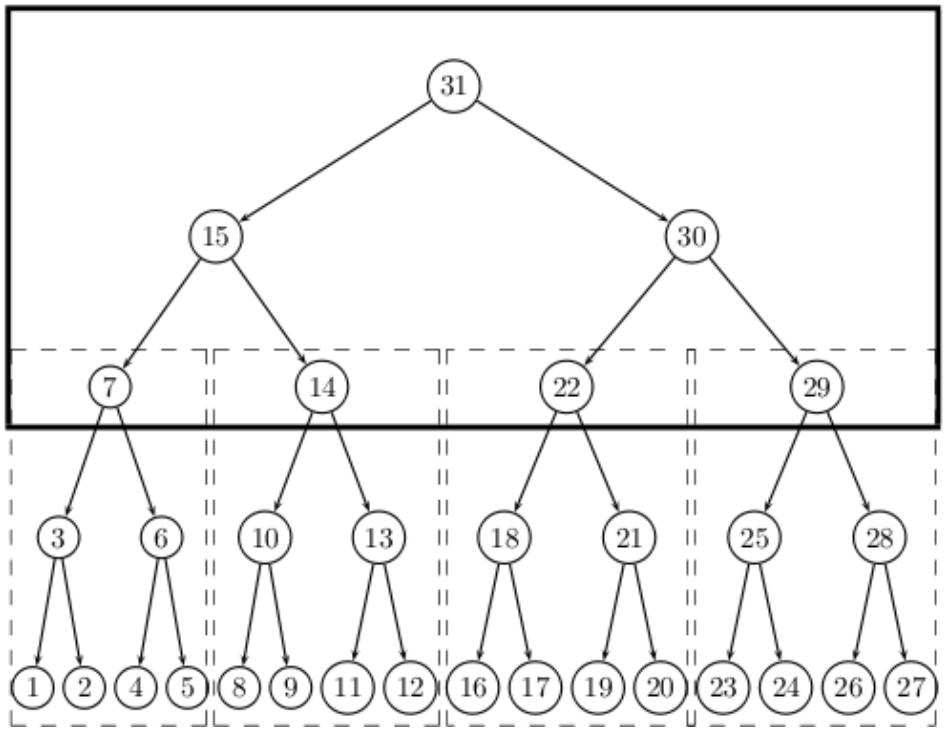
Parallelization (model problem)



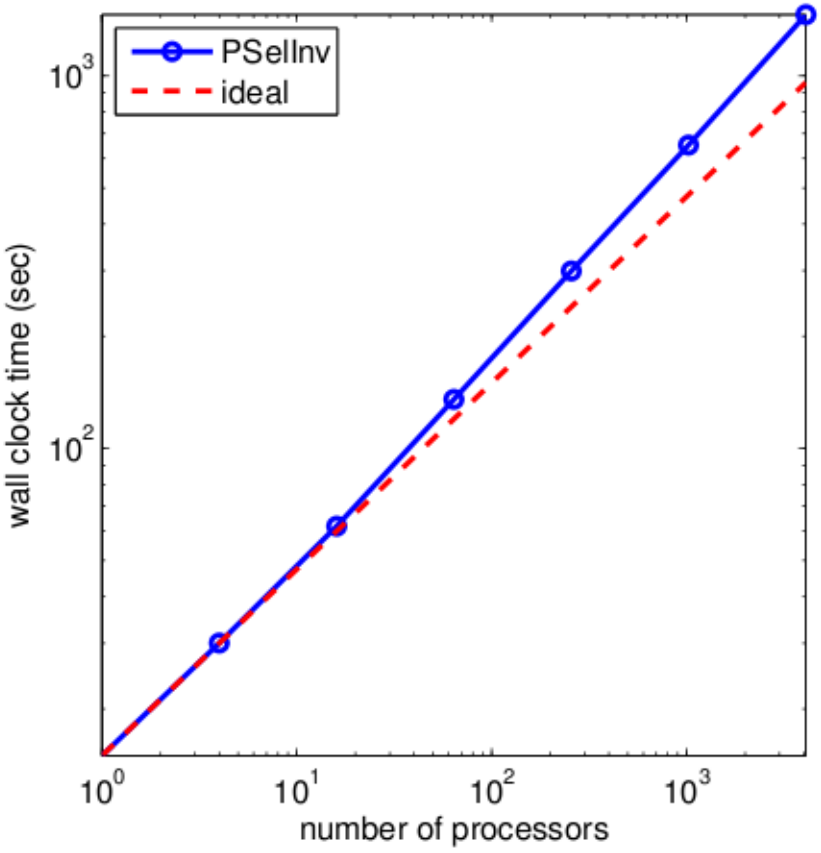


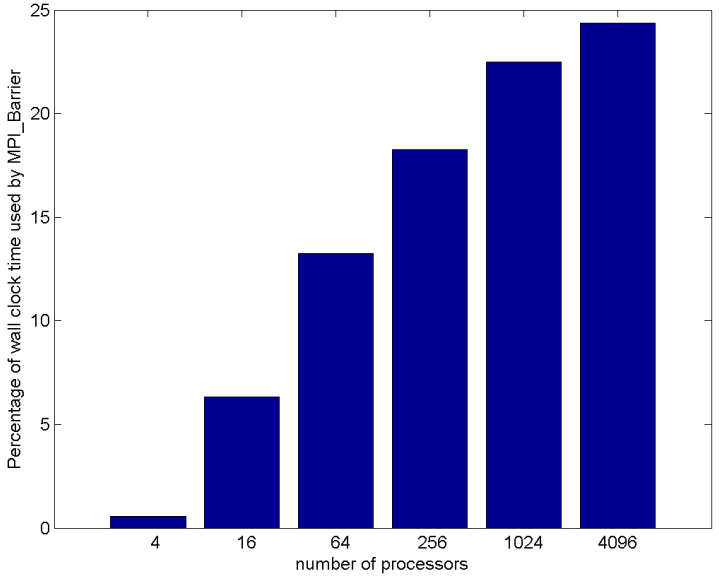
Task-to-processor map

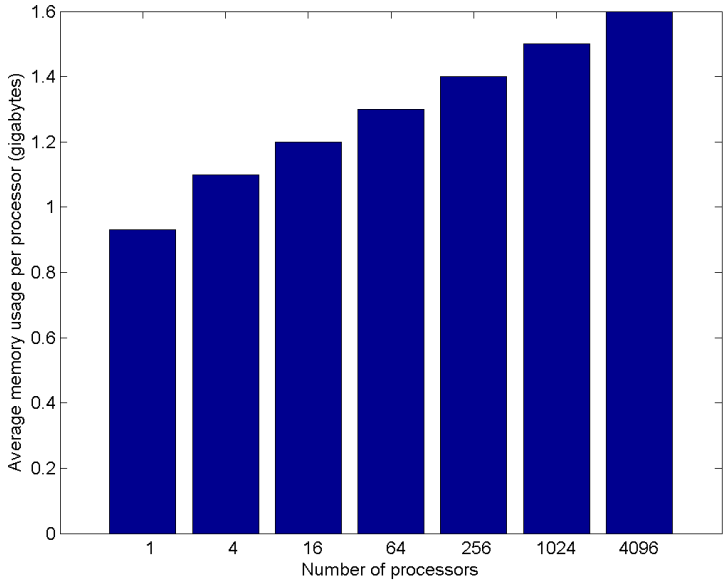




Performance







General problems

- Use SuperLU_DIST to factor (block fan-out)
- “Broadcast” a supernode column of L
- Restricted matrix-matrix multiplication
- “Reduce” onto a supernode column
- Overlap communication with matrix update