

Quantum Molecular Dynamics: Representation & Solution

Aiichiro Nakano

Collaboratory for Advanced Computing & Simulations

Department of Computer Science

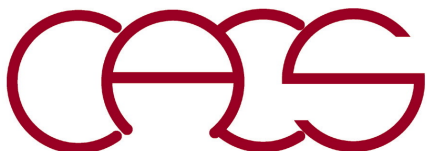
Department of Physics & Astronomy

Department of Quantitative & Computational Biology

University of Southern California

Email: anakano@usc.edu

How to represent & solve Kohn-Sham equations in QMD?

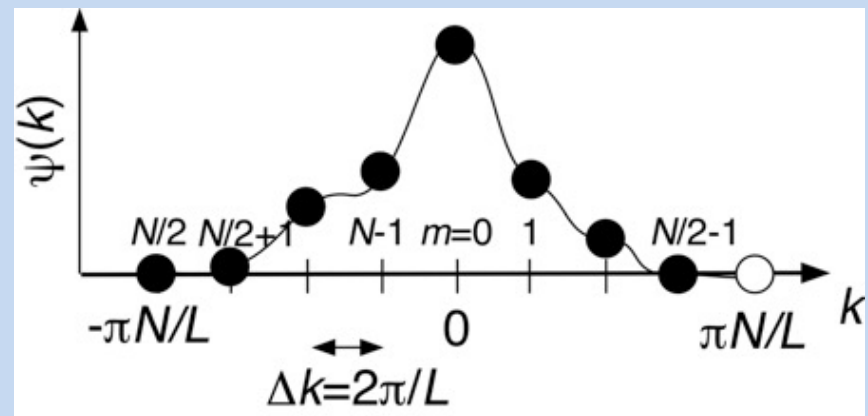
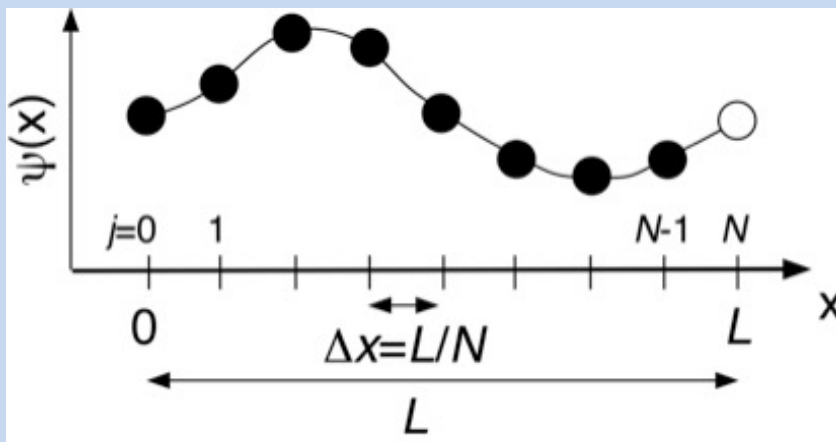


Representation: Plane-Wave Basis

- Pseudopotentials result in slowly varying wave functions that can be represented on a regular grid, which in turn can be represented as a linear combination of plane waves, *i.e.*, Fourier transform

$$\psi(\mathbf{r}_j) = \sum_{\mathbf{k}_n} \psi_{\mathbf{k}_n} \exp(i\mathbf{k}_n \cdot \mathbf{r}_j)$$

1D example

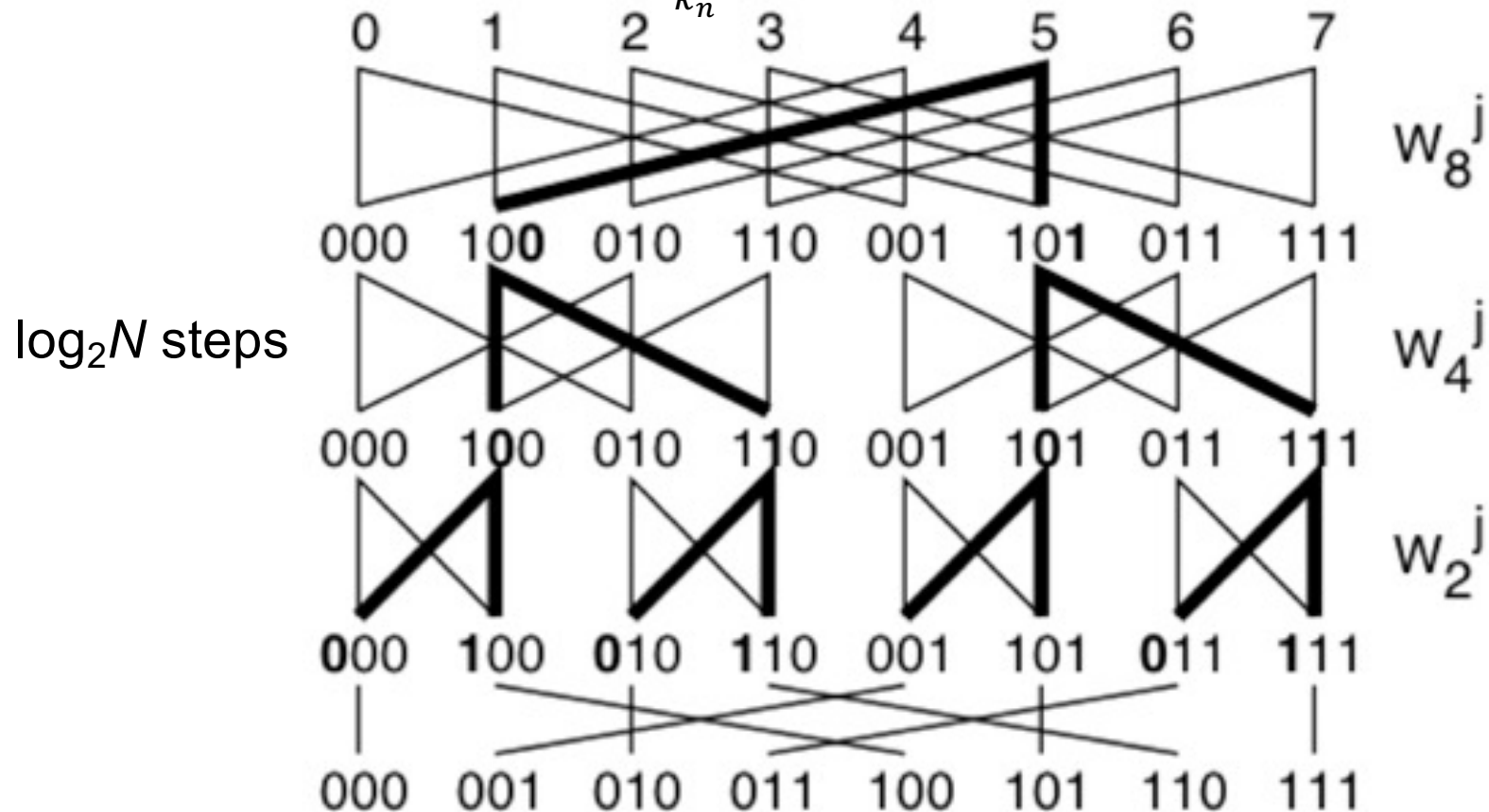


$$x_j = \frac{jL}{N}; \quad k_n = \frac{2\pi n}{L}$$

Numerics: Fast Fourier Transform

- $O(M \log N)$ fast Fourier-transform (FFT) algorithm is typically used to perform Fourier transform

$$\psi(x_j) = \sum_{k_n} \psi_{k_n} \exp(ik_n x_j)$$



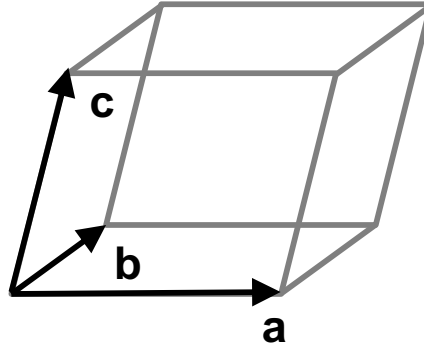
Butterfly (hypercube) data-exchange network

See PHYS 516 lecture on quantum dynamics

<https://aiichironakano.github.io/phys516/03QD-slide.pdf>

Periodic Solid

- Consider a periodic solid with the unit cell spanned by vectors \mathbf{a} , \mathbf{b} & \mathbf{c}



- Fourier transform of a periodic function

$$u(\mathbf{r}) = \sum_{\mathbf{G}} u_{\mathbf{G}} \exp(i\mathbf{G} \cdot \mathbf{r})$$

$$\mathbf{G} = \frac{2\pi}{\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})} [l(\mathbf{b} \times \mathbf{c}), m(\mathbf{c} \times \mathbf{a}), n(\mathbf{a} \times \mathbf{b})] \quad (l, m, n \in \mathbb{Z})$$

- Bloch's theorem

$$\begin{aligned} \psi_{n\mathbf{k}}(\mathbf{r}) &= \exp(i\mathbf{k} \cdot \mathbf{r}) u_{n,\mathbf{k}}(\mathbf{r}) \\ &= \sum_{\mathbf{G}} u_{n,\mathbf{k}}(\mathbf{G}) \exp(i(\mathbf{k} + \mathbf{G}) \cdot \mathbf{r}) \end{aligned}$$

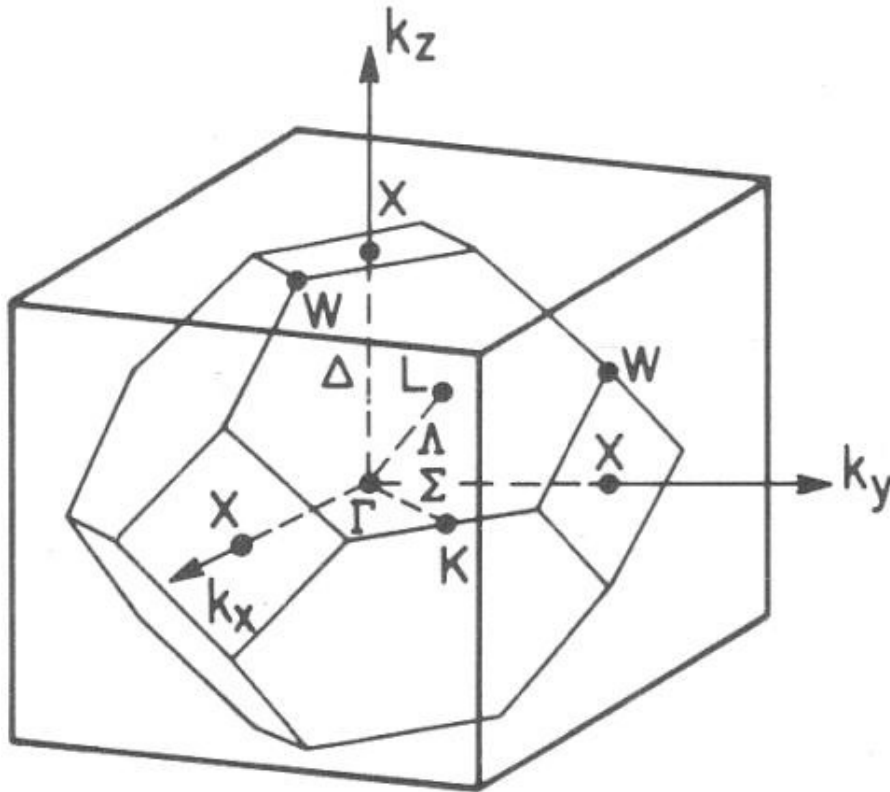
$\mathbf{k} \in$ first Brillouin zone in the reciprocal space

Electronic Bands: Infinite Lattice

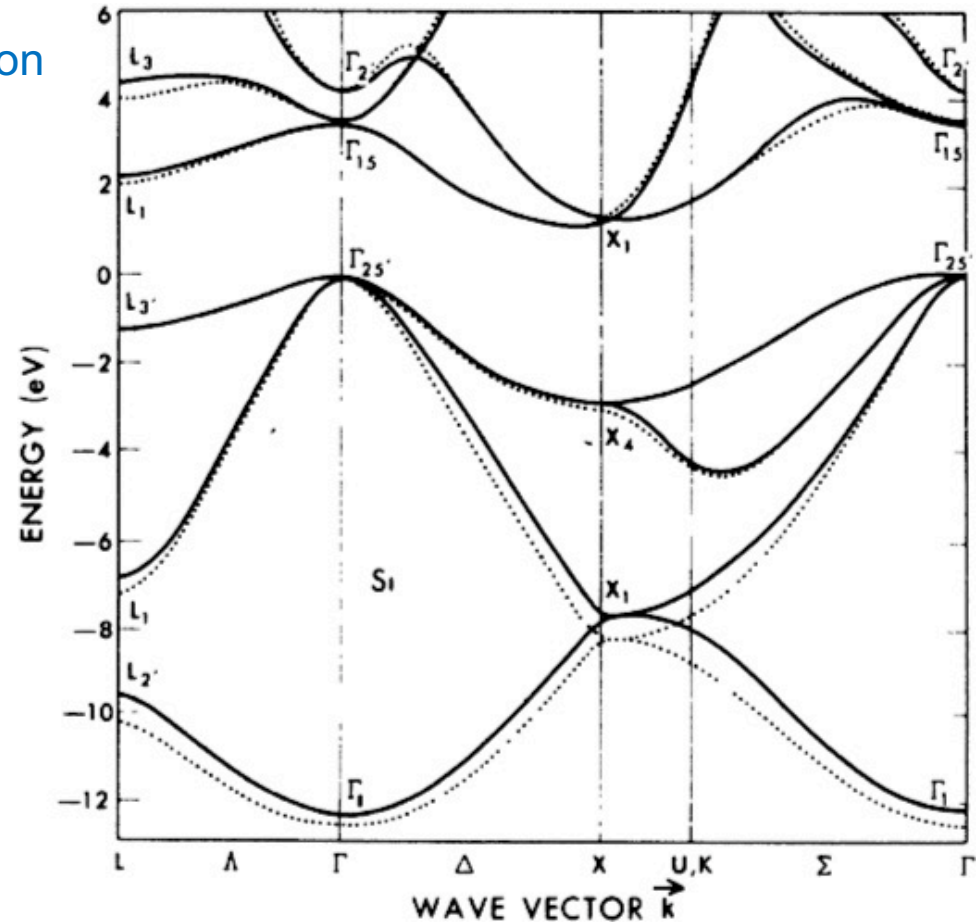
- **Bloch theorem:** $\psi_{n\mathbf{k}}(\mathbf{r}) = \exp(i\mathbf{k} \cdot \mathbf{r})u_{n,\mathbf{k}}(\mathbf{r})$

band index

periodic function



Brillouin zone of Si crystal



Kohn-Sham energy

J. R. Chelikowsky & M. L. Cohen, *Phys. Rev. B* 10, 5095 ('74)

See notes on (1) [plane-wave basis](#) & (2) [supercell](#)

QMD Algorithm

time $t = 0$

coordinates & velocities of atoms, $\{\vec{R}_I(t)\}, \{\vec{V}_I(t)\}$

minimize energy functional $E[\{\psi_n\}, \{\vec{R}_I(t)\}]$
by **conjugate-gradient method**

$$\text{atomic force } \vec{F}_I(t) = - \frac{\partial E[\{\psi_n\}, \{\vec{R}_I(t)\}]}{\partial \vec{R}_I(t)}$$

new coordinates and velocities of atoms at time $t + \Delta t$

$$\{\vec{R}_I(t + \Delta t)\}, \{\vec{V}_I(t + \Delta t)\}$$

by integrating **Newton's equations of motion**

$$M_I \frac{d^2 \vec{R}_I(t)}{dt^2} = \vec{F}_I(t)$$

$t \leftarrow t + \Delta t$

Molecular Dynamics Modes

- **Structural optimization, e.g., quasi-Newton method, see**
<https://aiichironakano.github.io/phys760/MNK.pdf>

> **Relax atomic positions $\{\mathbf{R}_I\}$ to minimize the energy**
 $\{\mathbf{R}_I^*\} = \operatorname{argmin}_{\{\mathbf{R}_I\}} \left(\min_{\{\psi_n(\mathbf{r})\}} E[\{\psi_n(\mathbf{r})\}, \{\mathbf{R}_I\}] \right)$

- **Molecular dynamics**

> **Follow atomic trajectories by numerically integrating Newton's second law of motion**

$$M \frac{d^2}{dt^2} \mathbf{R}_I = - \left\langle \Psi_0 \left| \frac{\partial h(\mathbf{r}, \mathbf{R}(t), t)}{\partial \mathbf{R}(t)} \right| \Psi_0 \right\rangle$$

> **Microcanonical (NVE), canonical (NVT) & isobaric (NPT) ensembles are supported**

Martyna *et al.*, *Mol. Phys.* **87**, 1117 ('96)

For classical molecular dynamics, see PHYS 516 (*Methods of Computational Physics*, <https://aiichironakano.github.io/phys516/02MD-slide.pdf>) & MASC 575 (*Basics of Atomistic Simulation of Materials*)

Self-Consistent Field Iteration

$$\left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial \mathbf{r}^2} + \hat{V}_{\text{ion}} + \hat{V}_{\text{H,xc}}[\rho(\mathbf{r})] \right) \psi_n(\mathbf{r}) = \epsilon_n \psi_n(\mathbf{r})$$

Given $\rho(\mathbf{r})$,
iteratively obtain
 $\{\psi_n, \epsilon_n\}$, e.g., by
preconditioned
conjugate gradient

Given $\{\psi_n, \epsilon_n\}$,
determine μ and
compute $\rho(\mathbf{r})$

$$\rho(\mathbf{r}) = \sum_n |\psi_n(\mathbf{r})|^2 \Theta(\mu - \epsilon_n)$$

Chemical potential

$$N = \int d\mathbf{r} \rho(\mathbf{r})$$

Self-Consistent Field Iteration

initial wave function $\{\psi_n | n = 1, \dots, N_{\text{band}}\}$ & charge ρ

solve $\nabla^2 V_{\text{Hartree}}(\vec{x}) = -4\pi e^2 \rho(\vec{x})$
set up the electronic potential, $V = V_{\text{ion}} + V_{\text{Hartree}} + V_{\text{xc}}$

unitary transformation to diagonalize $\int d^3x \psi_m(\vec{x}) \left\{ -\frac{\hbar^2}{2m} \nabla^2 + V(\vec{x}) \right\} \psi_n(\vec{x})$

iterative improvement of $\{\psi_n\}$ & orthonormalization

calculate new ρ from updated $\{\psi_n\}$

if not converged



```
graph TD; A["initial wave function {ψn | n = 1, ..., Nband} & charge ρ"] --> B["solve ∇²VHartree(x̄) = -4πe²ρ(x̄)  
set up the electronic potential, V = Vion + VHartree + Vxc"]; B --> C["unitary transformation to diagonalize ∫ d³x ψm(x̄) { -ħ²/2m ∇² + V(x̄) } ψn(x̄)"]; C --> D["iterative improvement of {ψn} & orthonormalization"]; D --> E["calculate new ρ from updated {ψn}"]; E -- "if not converged" --> B;
```

Orthogonalization by Matrix Decomposition

- **Gram-Schmidt orthonormalization:** The orthonormal basis set $Q = [q_1 \dots q_m]$ is obtained starting from an arbitrary set of m vectors, $S = [s_1 \dots s_m]$ as

$$\begin{aligned}
 & \mathbf{q}_1 = \mathbf{s}_1 / \|\mathbf{s}_1\| \\
 & \text{for } i = 2 \text{ to } m \\
 & \quad \mathbf{q}'_i = \mathbf{s}_i - \sum_{j=1}^{i-1} \mathbf{q}_j (\mathbf{q}_j \cdot \mathbf{s}_i) \quad \text{Projection!} \\
 & \quad \mathbf{q}_i = \mathbf{q}'_i / \|\mathbf{q}'_i\| \\
 & \text{endfor}
 \end{aligned}$$

$\hat{P} \quad |s_i\rangle$
 $\sum_{j=1}^{i-1} |\tilde{q}_j\rangle \langle q_j|$

- The Gram-Schmidt procedure amounts to QR decomposition, $S = QR$, where R is an $m \times m$ right-triangle matrix

$$\begin{aligned}
 & \begin{matrix} n & m \\ \left[\begin{array}{cccc} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 \end{array} \right] \end{matrix} = \begin{matrix} n & m \\ \left[\begin{array}{cccc} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 & \mathbf{q}_4 \end{array} \right] \end{matrix} \begin{matrix} m \\ \left[\begin{array}{cccc} \|\mathbf{q}'_1\| & \mathbf{q}_1 \cdot \mathbf{s}_2 & \mathbf{q}_1 \cdot \mathbf{s}_3 & \mathbf{q}_1 \cdot \mathbf{s}_4 \\ 0 & \|\mathbf{q}'_2\| & \mathbf{q}_2 \cdot \mathbf{s}_3 & \mathbf{q}_2 \cdot \mathbf{s}_4 \\ 0 & 0 & \|\mathbf{q}'_3\| & \mathbf{q}_3 \cdot \mathbf{s}_4 \\ 0 & 0 & 0 & \|\mathbf{q}'_4\| \end{array} \right] \end{matrix} \\
 & \quad \therefore \mathbf{s}_i = \|\mathbf{q}'_i\| \mathbf{q}_i + \sum_{j=1}^{i-1} \mathbf{q}_j (\mathbf{q}_j \cdot \mathbf{s}_i)
 \end{aligned}$$

- For higher parallelization, Cholesky decomposition is used instead

Charge Mixing

- **Fixed-point charge mapping in self-consistent field iteration**

$$\rho_{\text{in}}(\mathbf{r}) \mapsto v_{\text{Hxc}}(\mathbf{r}) \mapsto \{\psi_n(\mathbf{r})\} \mapsto \rho_{\text{out}}(\mathbf{r})$$

- **Directly using $\rho_{\text{out}}(\mathbf{r})$ as $\rho_{\text{in}}(\mathbf{r})$ in the next iteration step often destabilizes numerical iteration**

- **Charge mixing**

$$\rho_{\text{in}}^i \leftarrow \sum_{j=1}^n \alpha_j \rho_{\text{in}}^{i-j}$$

- **Determine the mixing coefficients α_i in order to minimize the residual**

$$R[\rho_{\text{in}}(\mathbf{r})] \equiv \rho_{\text{out}}[\rho_{\text{in}}] - \rho_{\text{in}}$$

- **See note on [Pulay charge mixing](#)**

Conjugate-Gradient Minimization of Energy Functional

i : iteration index; n : band index

“gradient”
$$g_n^{(i)} = - \frac{\delta E \left[\left\{ \psi_n^{(i)} \right\}, \left\{ \vec{R}_I(t) \right\} \right]}{\delta \psi_n^{(i)}} + \epsilon_n^{(i)} \psi_n^{(i)} \equiv -H \psi_n^{(i)} + \epsilon_n^{(i)} \psi_n^{(i)}$$

$$\epsilon_n^{(i)} = \int d^3 r \psi_n^{(i)*} H \psi_n^{(i)}$$

“preconditioning”
$$\tilde{g}_n^{(i)} = \hat{P} g_n^{(i)}$$

“conjugate gradient”
$$h_n^{(i)} = \tilde{g}_n^{(i)} + \beta h_n^{(i-1)}, \beta = \int d^3 r g_n^{(i)} \cdot g_n^{(i)} / \int d^3 r g_n^{(i-1)} \cdot g_n^{(i-1)}$$

“new wave function”
$$\psi_n^{(i+1)} = C(\lambda) \left(\psi_n^{(i)} + \lambda h_n^{(i)} \right)$$

 with constraint
$$\int d^3 r \psi_n^{(i+1)*} \psi_m = 0 \quad (m \leq n)$$

$$i \leftarrow i + 1 \quad \text{if } \left| \epsilon_n^{(i+1)} - \epsilon_n^{(i)} \right| > \varepsilon$$

See lecture on iterative minimization (<https://aiichironakano.github.io/phys516/QD2CG.pdf>) & notes on (1) conjugate-gradient (CG) method, (2) CG electronic-state solver, (3) CG DFT solver & (4) 2D electron example

Real-Space Grid as a Basis

- Wave functions & electron density are represented by numerical values on **real-space grid points**
- **Finite difference** expansion for the kinetic-energy operator

$$\left. \frac{\partial^2 \psi_n}{\partial x^2} \right|_{\mathbf{r}_{ijk}=(x_i, y_j, z_k)} = \sum_{l=-L}^L C_l \psi_n(x_i + lh, y_j, z_k) + O(h^{2L+2})$$

(short-ranged operation)

The calculations are performed completely in “real space”



- Suitable for systems with vacuum (*e.g.*, clusters, surfaces)
- Efficient implementation on parallel computers

Acceleration of Convergence

Preconditioning

Enhanced convergence rate of **short** wavelength components of the residual

$$(H - \varepsilon)(\psi + \delta) = 0$$

$$\rightarrow (H - \varepsilon)\psi = -g$$

$$(H - \varepsilon)\delta = g$$

approximately invert?
 $\delta \leftarrow (H - \varepsilon)^{-1}g$

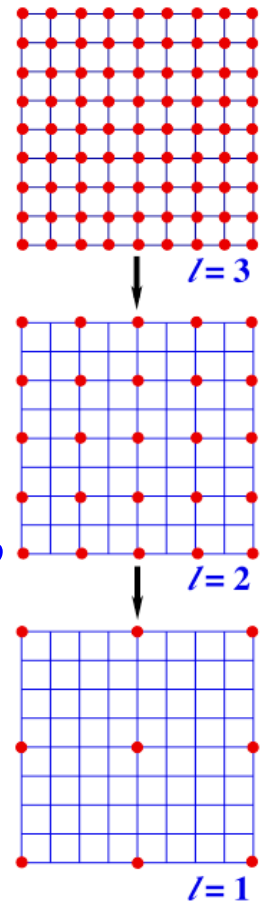
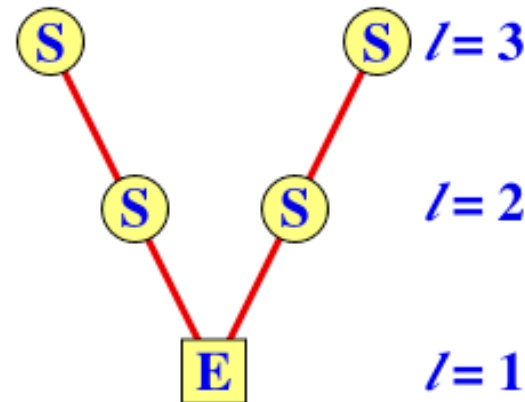
$$\begin{aligned} \tilde{g}_n(x_i, y_j, z_k) &= \hat{P} g_n^{(i)} \\ &= \sum_{l_1=-1}^1 \sum_{l_2=-1}^1 \sum_{l_3=-1}^1 c_{l_1 l_2 l_3} g_n(x_i + l_1 h, y_j + l_2 h, z_k + l_3 h) \end{aligned}$$

Multigrid method [Brandt '77, Bernholc *et al.* '96, Beck, '00]

To reduce **long** wavelength components of the residual,

$$\begin{aligned} \left(-\frac{\hbar^2}{2m} \nabla^2 + V \right) \varphi &= g_n^{(i)} \\ \psi_n^{(i)} &\leftarrow \psi_n^{(i)} + \varphi \end{aligned}$$

on a coarse grid



Iterative Solution of Linear Systems

$$Ax = b$$

$$A = \begin{matrix} & & L & & \\ & & & & \\ & & & & \end{matrix} + \begin{matrix} & & D & & \\ & & & & \\ & & & & \end{matrix} + \begin{matrix} & & U & & \\ & & & & \\ & & & & \end{matrix}$$

$$\begin{bmatrix} X & & & & \\ X & X & & & \\ X & X & X & & \end{bmatrix} + \begin{bmatrix} X & & & & \\ & X & & & \\ & & X & & \\ & & & X & \\ & & & & X \end{bmatrix} + \begin{bmatrix} & & X & & \\ & & & X & \\ & & & & X \\ & & & & \\ & & & & \end{bmatrix}$$

- **Fixed-point equation**

$$x = D^{-1}[-(L+U)x + b]$$

- **Jacobi iteration**

$$x^{(n+1)} = D^{-1}[-(L+U)x^{(n)} + b]$$

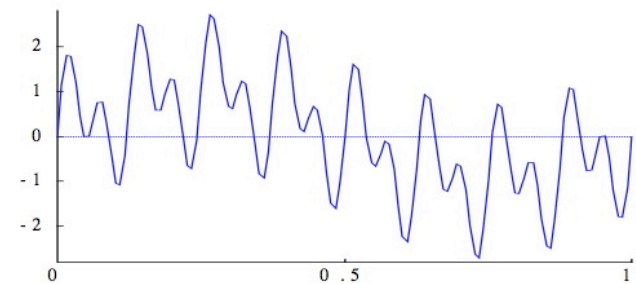
$$x_i^{(n+1)} = \frac{1}{a_{ii}} \left(- \sum_{\substack{j=1 \\ (j \neq i)}}^N a_{ij} x_j^{(n)} + b_i \right)$$

- **Over (under) relaxation: $\Delta > 1$ ($\Delta < 1$)**

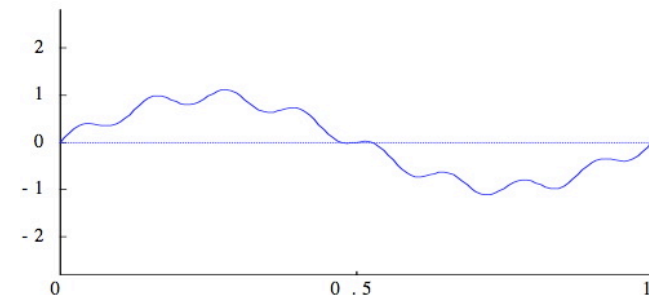
$$x_i^{(n+1)} = \frac{1}{a_{ii}} \left(- \sum_{\substack{j=1 \\ (j \neq i)}}^N a_{ij} x_j^{(n)} + b_i \right)$$

$$= x_i^{(n)} + \frac{1}{a_{ii}} \left(- \sum_{j=1}^N a_{ij} x_j^{(n)} + b_i \right) \rightarrow x_i^{(n)} + \frac{\Delta}{a_{ii}} \left(- \sum_{j=1}^N a_{ij} x_j^{(n)} + b_i \right)$$

- **Initial error:**



- **Error after 35 iteration sweeps:**



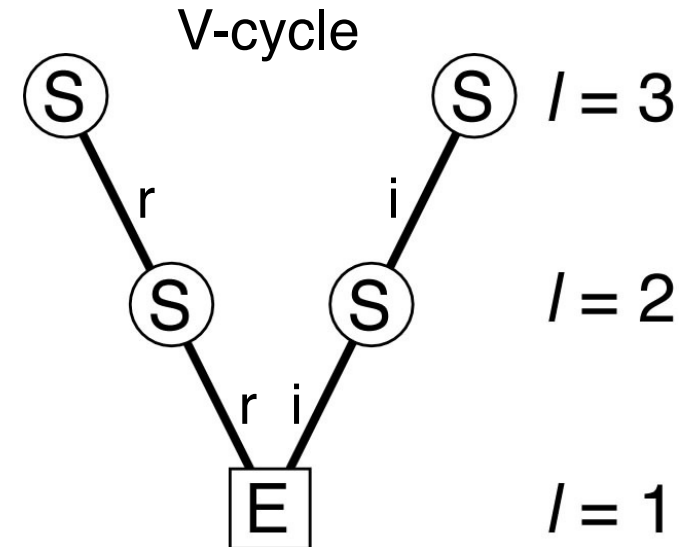
Multigrid Method

- Residual equation:** $A^{(l)}(\psi + e) = 0$

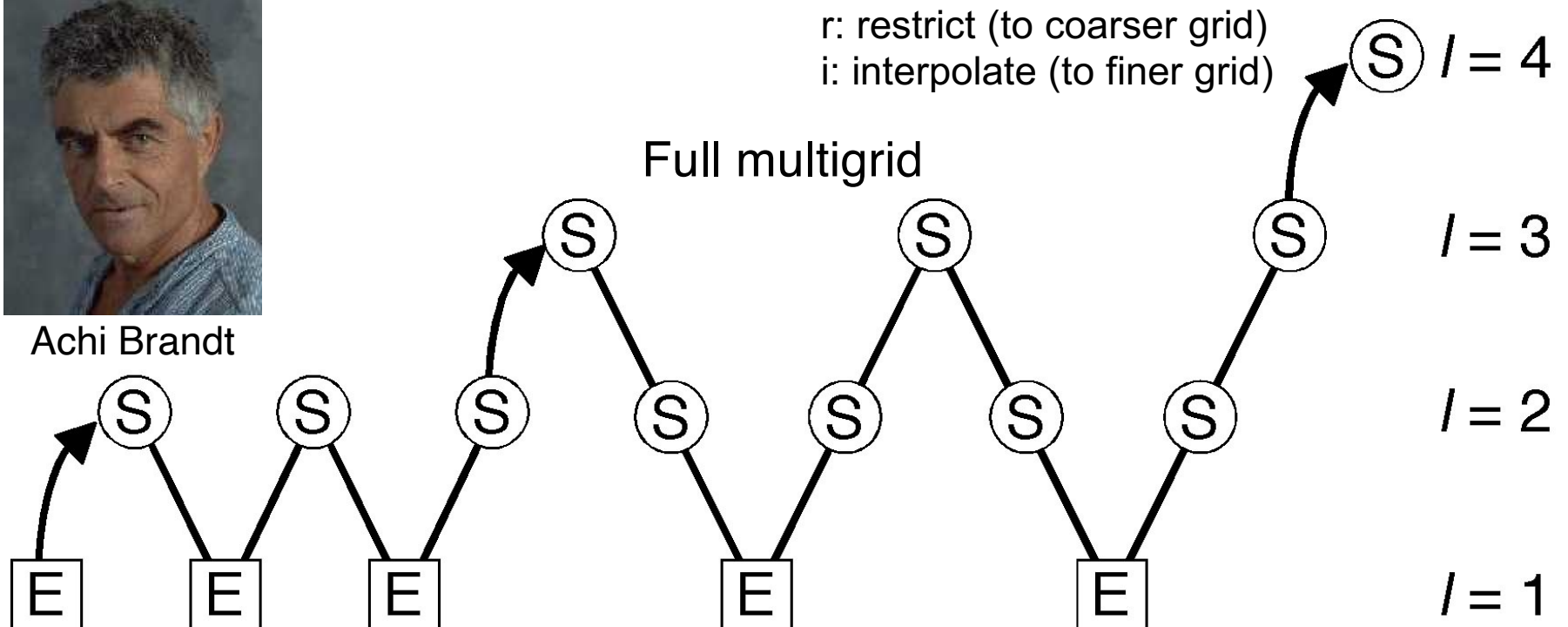
$$\frac{-)A^{(l)}\psi = r}{A^{(l)}e = -r}$$

- Smoothing:** $e \leftarrow [1 + Z^{(l)}A^{(l)}]e + Z^{(l)}r$

- Coarsening of residual & interpolation of error**

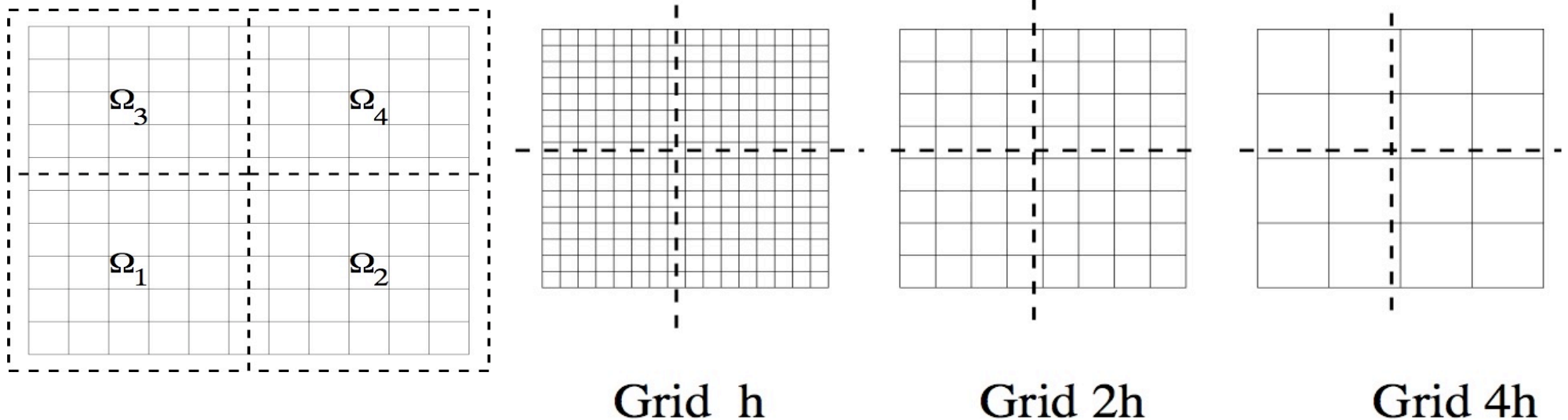


Achi Brandt



Parallel Multigrid Method

- Domain decomposition with boundary-layer caching



- 2D computational & communication costs (isogranular or weak scaling)

$N \times N$ grids each on $P \times P$ processors: $T(N^2 P^2, P^2) = a \log NP + bN + cN^2$

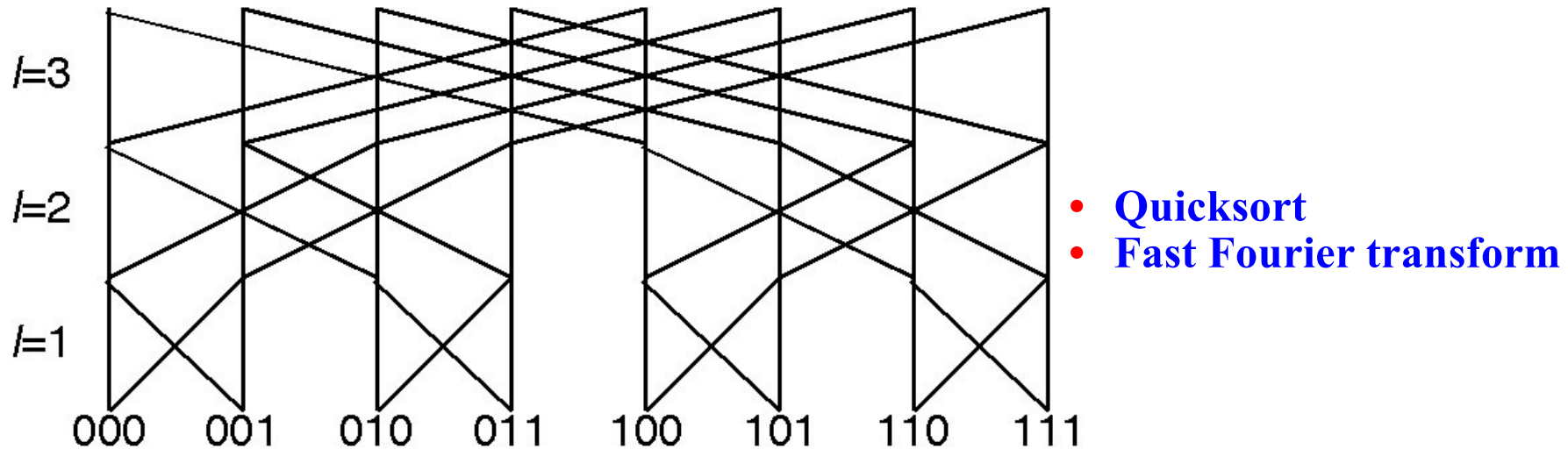
$$S_{P^2} = \frac{N^2 P^2 T(N^2, 1)}{N^2 T(N^2, P^2)} = \frac{P^2 (cN^2)}{a \log NP + bN + cN^2} = \frac{P^2}{1 + \frac{b}{cN} + \frac{a}{cN} \log NP}$$

$$E_{P^2} = \frac{S_{P^2}}{P^2} = \frac{1}{1 + \frac{b}{cN} + \frac{a}{cN} \log NP}$$

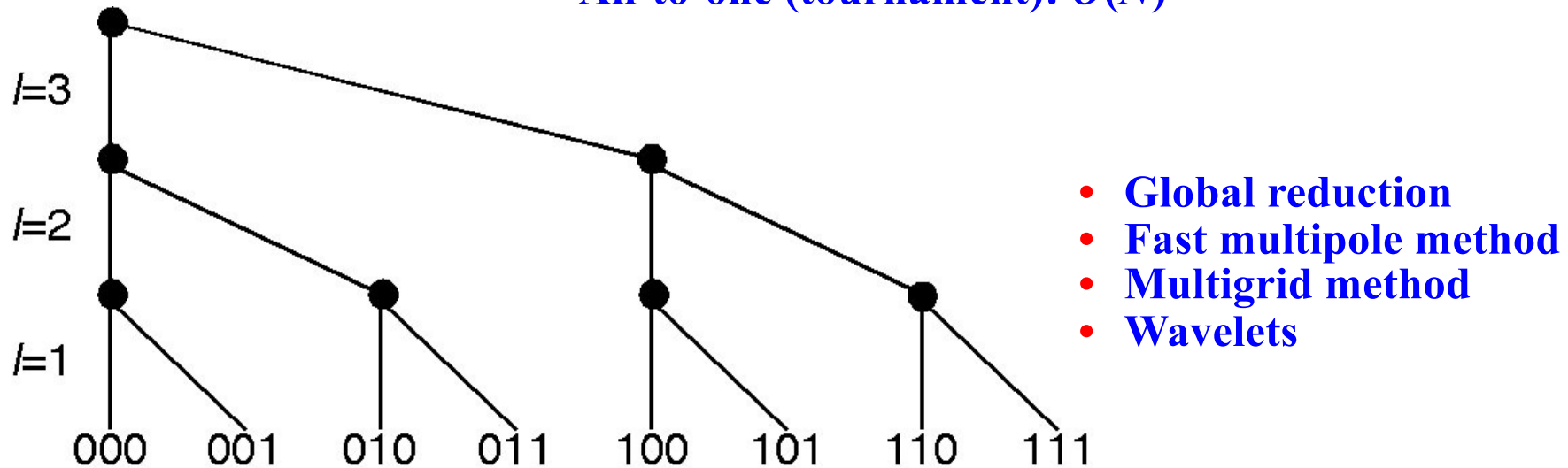
Nakano *et al.*, *Comput. Phys. Commun.* **83**, 181 ('94)

Global Communications

All-to-all (hypercube): $O(N \log N)$



All-to-one (tournament): $O(N)$

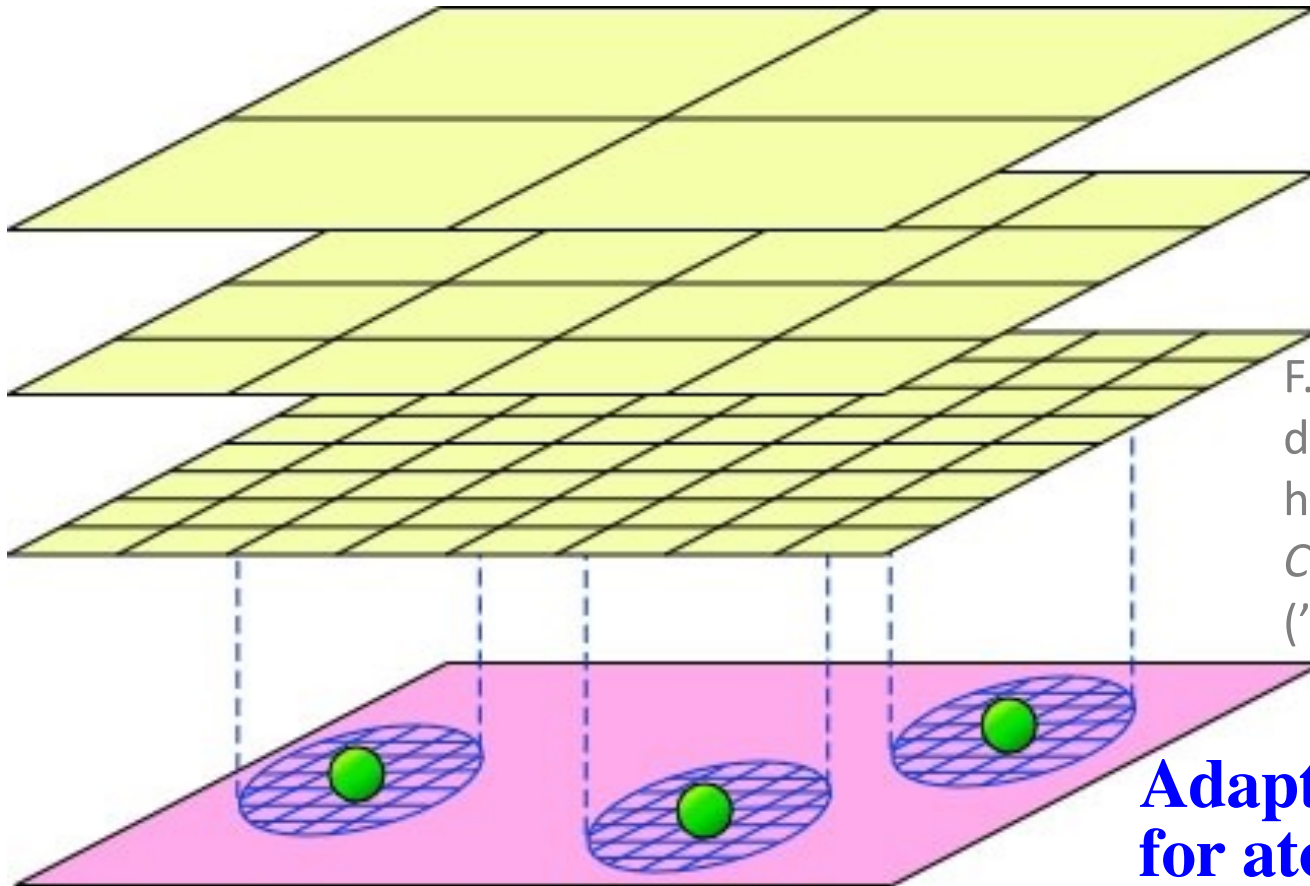


See note on [multigrid preconditioned CG](#)

Real-Space DFT on Hierarchical Grids

Efficient parallelization of DFT: real-space approaches

- **High-order finite difference** [Chelikowsky, Troullier, Saad, '94]
- **Multigrid acceleration** [Bernholc *et al.*, '96; Beck, '00]
- **Double-grid method** [Ono, Hirose, '99] ~ **obsolete, with PAW**
- **Spatial decomposition/divide-&-conquer**



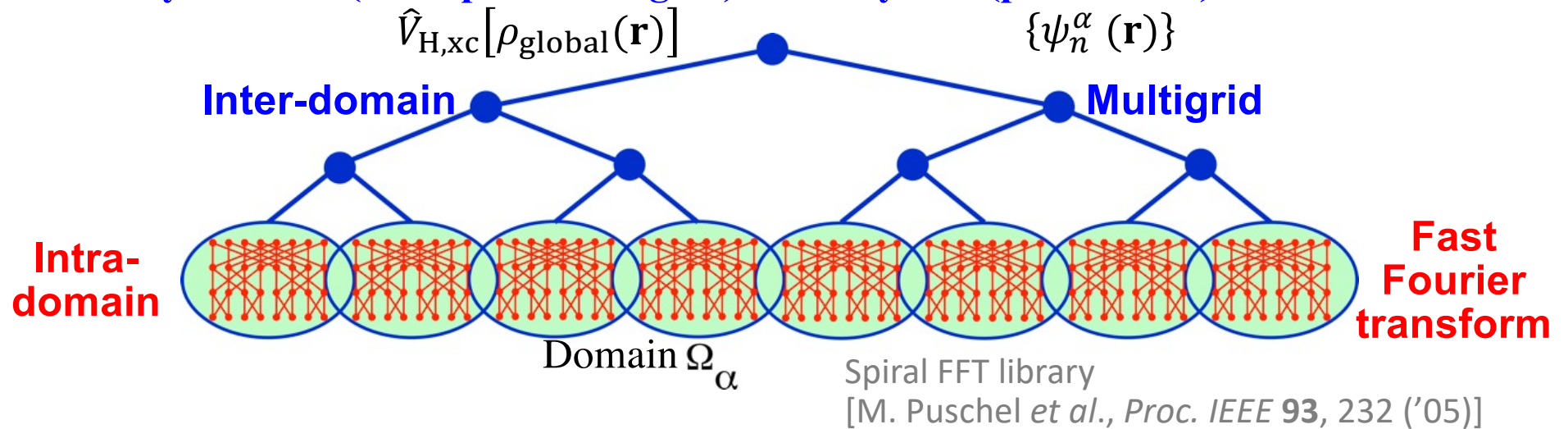
**Multigrid acceleration
of preconditioned
conjugate gradient**

F. Shimojo *et al.*, "Embedded
divide-and-conquer algorithm on
hierarchical real-space grids,"
Comput. Phys. Commun. **167**, 151
(05)

**Adaptive high-resolution grid
for atomic pseudopotentials**

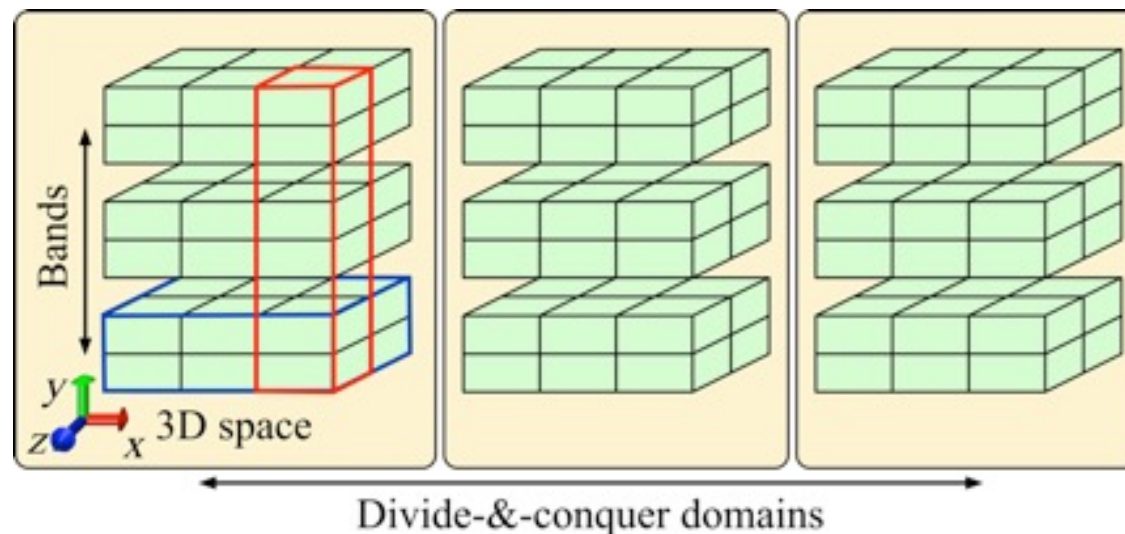
Hierarchical Computing

- Globally scalable (real-space multigrid) + locally fast (plane wave) electronic solver



cf. globally- sparse-yet-locally-dense eigensolver [J. H. Lam et al., *Nature Commun.* **15**, 3479 ('24)]

- Hierarchical band (*i.e.*, Kohn-Sham orbital) + space + domain (BSD) decomposition



Finite-Element DFT

- DFT calculation using a higher-order adaptive spectral finite-element (FE) basis outperforms that with the plane-wave basis for larger (e.g., > 10,000 electrons) systems: see DFT-FE code [S. Das *et al.*, *Comput. Phys. Commun.* **280**, 108473 ('22); <https://github.com/dftfeDevelopers/dftfe>]
- 2023 Gordon-Bell award: 659.7 Pflop/s (43.1% of the peak) by the DFT-FE code for 619,124 electrons on 8,000 GPU nodes of the Frontier supercomputer [S. Das *et al.*, *Proc. Supercomputing, SC* ('23)]

