Conquest
— order N ab initio Electronic Structure simulation code for quantum mechanical modelling in large scale

Lianheng Tong
Overview

• Overview of Conquest project
• Brief Introduction to DFT
• Structure of Conquest
• Parallelisation and Linear Scaling
• Applications
Conquest

- Density Functional Theory Code: simulating electronic structures of materials
- Linear Scaling
- Written in Fortran 90
- Pure MPI parallelisation at moment, OpenMP+MPI hybrid in works
- Available as BETA release on HPC platforms: HECToR, K-Computer, NIMS simulator
- Supported by dCSE, HECToR
- Website: http://www.order-n.org/
Developers

- David Bowler (UCL)
- Lianheng Tong (UCL)
- Conn O'Rourke (UCL)
- Umberto Terranova (UCL)
- Veronika Brazdova (UCL)
- Mike Gillan (UCL)
- Lionel Truflandier (Bordeaux, France)
- Tsuyoshi Miyazaki (NIMS, Japan)
- Michiaki Arita (NIMS, Japan)
Density Functional Theory

- Nano-scale: Quantum Mechanics becomes important.
- Solving Schrödinger equation
  \[ \hat{H}\psi(r) = E\psi(r) \]
- Multiparticle wavefunction approach: Number of parameters: \( M = p^{3N}, \quad 3 \leq p \leq 10 \)
Density Functional Theory

• To find Ground state energy, instead of solving

\[ \hat{H} \psi(r) = E \psi(r) \]

• We write:

\[
E_{KS} = E_K[\rho(r)] + E_H[\rho(r)] + E_{xc}[\rho(r)] + E_{\text{pseudo}}([\rho(r)], R_i) + E_{\text{ion}}(R_i, R_j)
\]

and minimise the total energy with respect to density \( \rho(r) \).

• Practical to work with Hamiltonian

\[
H = -\frac{1}{2} \nabla^2 + V_H + V_{xc} + V_{\text{pseudo}}, \quad V_X = \frac{\delta E_X}{\delta \rho(r)}
\]

and minimise \( E = \text{tr}(\rho H) \)
Conquest

- We use density matrix
  \[ \rho(r_1, r_2) = \sum_n f_n \psi_n(r_1) \psi_n^*(r_2), \quad \rho(r) = \rho(r, r) \]
- We work directly with it using support functions
  \[ \rho(r_1, r_2) = \sum_{i \alpha j \beta} \phi_{i \alpha}(r_1) K^{i \alpha j \beta} \phi_{j \beta}(r_2) \]
- Locality imposed via radius of support functions and cutoff on \( K^{i \alpha j \beta} \)
- Support functions represented by basis functions
  \[ \phi_{i \alpha} = \sum_n C_{i \alpha n} \chi^n(r) \]
Basis Sets

- We now have two basis sets in Conquest:
  - Blips (piece-wise cubic splines defined on a grid moving with atoms)
    \[
    \Theta (\mathbf{r}) = \theta_0(x)\theta_0(y)\theta_0(z)
    \]
    \[
    \theta_0(x) = \begin{cases} 
    1 - \frac{3}{2}x^2 + \frac{3}{4}|x|^3 & \text{if } 0 < |x| < 1 \\
    \frac{1}{4}(2 - |x|)^3 & \text{if } 1 < |x| < 2 \\
    0 & \text{if } 2 < |x| 
    \end{cases}
    \]
  - Pseudo Atomic Orbitals, PAOs (radial term multiplied by spherical harmonic)
    - PAOs allow many analytic or local calculations (efficient)
    - Blips allow systematically improving basis set
Minimisation Procedures

• Three levels:
  • Innermost loop: minimise $E = \text{tr}(K H)$ w.r.t. $K$
  • Middle loop: seek self-consistent charge and potential
  • Outer loop: minimise energy w.r.t. basis set—vary $C_{i\alpha}^m$
Linear Scaling

- Normal DFT calculations: Cost is of order $N^3$ to $N^2$
- Large systems: needs linear ordering
- Localisation is key: everything must have finite range
- Density is local: $\rho(r_1, r_2) \to 0, \quad \|r_1 - r_2\| \to \infty$
- We impose: $\rho(r_1, r_2) = 0, \quad \|r_1 - r_2\| \geq R_{\text{cut}}$
- Interactions are ranged
Linear Scaling

- In practice each atom has a *halo*
- Support functions are *strictly local*
  \[ \phi_{i\alpha}(r) = 0, \| r - R_i \| \geq R_c^\phi \]
- *Only non-zero matrix elements are stored.*
- *Sparse matrix multiplication*

\[
L_{i\alpha j\beta} \neq 0 \\
L_{j\beta k\gamma} \neq 0 \\
R_{cut} = \max(|R_{i\text{halo}} - R_{j\text{halo}}|) \\
L_{i\alpha k\gamma} = 0
\]
Parallelisation

• There are three main areas in Conquest

  • Integration (on 3D integration/FFT grid)

  \[ H_{i\alpha j\beta} = \int d^3r \phi_{i\alpha}(r) \hat{H} \phi_{j\beta}(r) \]

  \[ S_{i\alpha j\beta} = \int d^3r \phi_{i\alpha}(r) \phi_{j\beta}(r) \]

• Matrix multiplication

• FFTs (on 3D integration/FFT grid)
Parallelisation: Integration/FFT grid

- Grid divided into blocks (for efficiency)
- Space divided into partitions, each partition is then allocated to a node
- Set of all blocks in charge by a node is called a domain
- A node stores value of all support functions at grid points within its domain
- Each node is responsible for partial contributions to all integrals (matrix elements) involving support functions touching its domain. Calculations are local
Parallelisation: Matrices

- Divides space into partitions, each partition has a set of atoms.
- Each node in charge of a set of partitions, and hence a set of atoms.
- Matrix distributed to nodes according to rows associated to the atoms in charge.
- Results of integrations need to be redistributed to follow matrix format.
- Matrix multiplication needs data fetch from other nodes.

\[
C^{i\alpha}_{j\beta} = \sum_{k\gamma} A^{i\alpha}_{k\gamma} B^{k\gamma}_{j\beta}
\]
Strong Scaling

- Ge hut cluster on Si(001) surface
- 11,620 atoms in unit cell
- Using between 16 to 512 cores

![Graph showing strong scaling for Ge hut cluster on Si(001) surface with 11,620 atoms in unit cell using between 16 and 512 cores. Efficiency is defined as actual speed up divided by increase in number of cores. The system demonstrates perfect linear scaling while time per core is invariant, indicating excellent parallelization.](image)

References:

Weak Scaling

- 512 Si atoms per core (memory limited)
- Four support functions per basis function, slightly coarse grid
- Self-consistency done for small cells
- Graph shows total time (sum of times on each core)
### Million Atoms DFT

<table>
<thead>
<tr>
<th>Atoms</th>
<th>Time/core (s)</th>
<th>Energy (Ha)</th>
<th>Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,096</td>
<td>7068.878</td>
<td>-308.268</td>
<td>8</td>
</tr>
<tr>
<td>32,768</td>
<td>6893.759</td>
<td>-2,466.150</td>
<td>64</td>
</tr>
<tr>
<td>262,144</td>
<td>6931.418</td>
<td>-19,729.202</td>
<td>512</td>
</tr>
<tr>
<td>2,097,152</td>
<td>7032.496</td>
<td>-157,833.618</td>
<td>4096</td>
</tr>
</tbody>
</table>

HECToR Phase 2a
- Dimyristoyl-sn-glycero-phosphocholine
- Phospholipid that incorporates choline as head group.
- Important part of cell membrane
- Hydrophilic heads in outer surfaces of the bilayer, hydrophobic in between.
- gramicidin A ion channel allows selective ions (mono-valency cations) to pass through cell membrane
- Over 15000 atoms, full DFT (SCF) calculation
DMPC-gA in Water

- Basis: Optimised SZP from DZP (using diagonalisation, piece-wise)
- XC Functional: PBE(+D2)
- Cores: 512, NIMS Simulator 1
- Intel Xeon processor Nehalem-EP (2.8 GHz), 4 cores/node, 2.85GB per core
- SCF computation time: < 24 Hrs
Summary

- Presented some details of Conquest
  - Linear scaling DFT code
  - Excellent scaling on HPC platforms
  - Capable of performing full quantum mechanical calculations on millions of atom
Summary

• On Fortran 90
  • Reasons we choose Fortran 90
    • HPC platforms supports Fortran or C
    • Compatibility with existing libraries
    • Array arithmetics
    • Modules, interfaces and also optional variables
  • Issues (personal experiences)
    • Some compilers can be too forgiving
    • Allows F77 syntax, which some times can lead to confusion
    • Question: automatic arrays vs. allocatable arrays—stack or heap?
Acknowledgements

- Conquest: David Bowler, Tsuyoshi Miyazaki, Mike Gillan
- Funding:
  - EPSRC
  - HECToR (NAG) dCSE
Thank You!