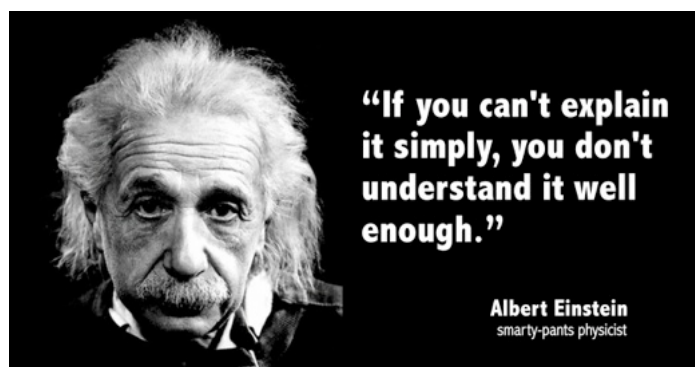


## Overview

- Background
- Classical Atomistic Simulation
- Essential Quantum Mechanics
- DFT: Approximations and Theory
- DFT: Implementation using plane waves and pseudopotentials
- Summary

Density Functional Theory For Dummies 2



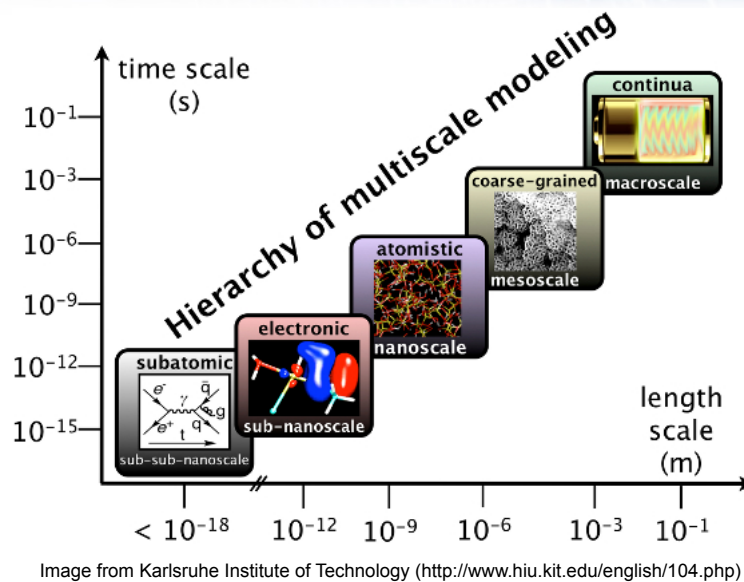
- Code Usage on HECToR (2011-12) by CPU Time:

Rank	Code	CPU Time	Method
1	VASP	3,436,163	DFT
2	CP2K	1,388,231	DFT
3	GROMACS	1,004,256	Classical
4	DL_POLY	800,829	Classical
5	CASTEP	684,149	DFT
9	ChemShell	313,364	DFT/ Classical
12	LAMMPS	238,586	Classical
13	ONETEP	157,370	DFT

- 53% of all CPU time used by Chemistry / Materials Science

## Classical Atomistic Simulation

epcc



Density Functional Theory For Dummies

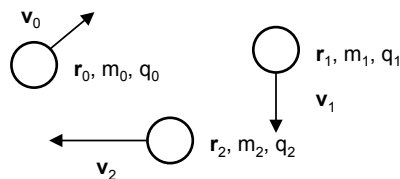
5

## Classical Atomistic Simulation

epcc

- The main elements of the simulation model are:

- Particles



- Force field

- Pair potentials
- Three-body
- Four-body
- e.g. CHARMM, GROMOS, AMBER, AMOEBA, ReaxFF ...

Density Functional Theory For Dummies

6

## Classical Atomistic Simulation



- Molecular Dynamics

- Newton's 2<sup>nd</sup> Law

$$F = ma$$

- Integrate using e.g. Velocity Verlet algorithm

$$r(t), \dot{r}(t) \rightarrow r(t + \delta t), \dot{r}(t + \delta t)$$

- Structural Optimisation

- Minimise total energy w.r.t. atomic positions

## Classical Atomistic Simulation



- Successes:

- Computationally cheap and parallelises well ( > 1,000,000,000 atoms on 10,000 CPUs)
- Able to predict mechanical properties
  - Density, elasticity, compressability, heat capacity (of insulators)
- Can predict structure
  - e.g. protein folding

- Failures:

- Anything involving electron transfer (i.e. all of Chemistry!)
  - Bonding, electrochemistry
  - Heat capacity of metals
  - Electronic structure/conductivity
  - Magnetic properties
  - etc.

## Essential Quantum Mechanics

- We need a model which can describe electrons...
- ... so turn to Quantum Mechanics – the Physics of the very small.
  - Thanks to Planck, Heisenberg, Einstein, Schrödinger *et al*, 1920s
- No longer think of point particles



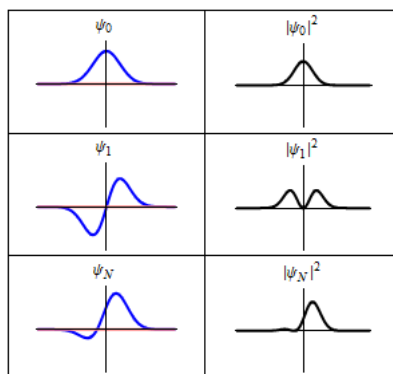
## Essential Quantum Mechanics

- Instead, particles are described by a wave-function:

$$\Psi(\vec{r}, t)$$

- No longer have a defined position but instead a probability distribution function:

$$|\Psi(\vec{r}, t)|^2$$



- The wavefunction captures *all* the observable behaviour of the particle.
- To compute a particular property (or observable) we apply the corresponding operator and extract the property as an eigenvalue (e.g. linear momentum):

$$\hat{P}_x \Psi(\vec{r}, t) \equiv -i\hbar \frac{\partial}{\partial x} \Psi(\vec{r}, t) = p_x \Psi(\vec{r}, t)$$

- The central equation(s) of Quantum Mechanics is(are) the Schrödinger's Equation(s)
- In the general, time-dependent form:

$$\hat{H}\Psi(\vec{r}, t) = i\hbar \frac{\partial}{\partial t} \Psi(\vec{r}, t)$$



- Wavefunctions which are 'stationary states' (the PDF is not time dependent) are described by the time-independent SE:

$$\hat{H}\Psi(\vec{r}) = E\Psi(\vec{r})$$

- For atomistic simulation, we typically have many atoms, each with many electrons, so the wavefunction depends on *all* the atomic and electronic coordinates (and time):

$$\Psi(\vec{R}_1, \vec{R}_2, \dots, \vec{R}_N, \vec{r}_1, \vec{r}_2, \dots, \vec{r}_n, t)$$

- This many-body wavefunction is a non-trivial (and unknown) coupling between all the particles in our system
- Too hard to deal with directly, so start making approximations...

- The Born-Oppenheimer approximation (1927)



- Nuclei are *much* more massive than electrons and move much slower (by  $\sim 10^3$ - $10^5$  times)
- So we can consider the nuclei to be fixed at some particular time  $t$  and solve the Time-Independent Schrödinger Equation for the electronic system in an external potential created by the nuclei:

$$\Psi(\vec{R}_1, \vec{R}_2, \dots, \vec{R}_N, \vec{r}_1, \vec{r}_2, \dots, \vec{r}_n, t)$$

↓

$$\Psi(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n)$$



- B-O Molecular Dynamics

- Solve the electronic system to get a total electronic energy as a function of the nuclear coordinates
- Then evolve the nuclear system with the electronic energy acting as a potential i.e.

$$E(\vec{R}_I) \rightarrow \vec{F}_I = \frac{\partial E(\vec{R}_I)}{\partial \vec{R}_I}$$

- And repeat...
- Can also apply all of our techniques from classical atomistic simulation to get structures, dynamics, mechanical properties ...
- Not to be confused with Car-Parrinello Method (CPMD)

- We need to solve the TISE to get the total energy of the systems but what ingredients go into our Hamiltonian?

- Kinetic energy (of the electrons – nuclei are stationary)
- Potential energy
  - of the electrons interacting with the nuclei
  - and electrons interacting with other electrons

$$\left[ -\frac{1}{2} \nabla^2 + \hat{V}_{ext}(\{\vec{R}_I\}, \{\vec{r}_i\}) + \hat{V}_{e-e}(\{\vec{r}_i\}) \right] \Psi(\{\vec{r}_i\}) = E \Psi(\{\vec{r}_i\})$$



## DFT: Approximations and Theory



- First attempt, guess many-particle wave function is a linear combination of products (Slater Determinant) of single-particle wave functions (Hartree & Fock, 1935)
  - First practical implementation in 1969 (Gaussian 70)
  - Led to Nobel Prize in Chemistry (1998) for John Pople
  - Computationally demanding  $\geq O(n^4)$
  - So limited to relatively small numbers of atoms (<100)
- H-F methods are often referred to as ‘wave function theory’
  - Even with the H-F ansatz, the many-particle wave function is still too complicated for practical use, so we make further simplifications...

## DFT: Approximations and Theory



- Hohenberg-Kohn Theorems, 1964
  - The external potential  $\hat{V}_{ext}(\{\vec{r}_i\})$  is a unique functional of the electron density  $n(\vec{r})$ 
    - i.e. if we know  $n(\vec{r})$  (a 3-dimensional function), we can calculate all the properties of the system as if we knew  $\Psi(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n)$ , a 3n-dimensional function!
  - A variational principle for electronic density exists:
$$E_v[n(\vec{r})] = T[n(\vec{r})] + U[n(\vec{r})] + V[n(\vec{r})], \quad E[\tilde{n}(\vec{r})] \geq E_0$$
    - The correct density gives the lowest Energy
    - So by iteratively improving our guess, we can approach the ‘correct’ density.

## DFT: Approximations and Theory

- One final step needed to compute these functionals...
- Kohn-Sham Method (1965) (Kohn shared NP with Pople)
  - Instead of the the many-body system of interacting electrons, define a set of 'KS-orbitals' (c.f. wavefuctions) of fictitious, non-interacting electrons moving in an effective potential:
 
$$V_{KS}(\vec{r}) = V_{ext}(\vec{r}) + V_H(n(\vec{r})) + V_{XC}(n(\vec{r}))$$
  - Then the orbitals which satisfy the K-S Equations (c.f. Schrödinger Eq) give the same density as the interacting system!
  - All the electron-electron interactions are included in the exchange-correlation potential
  - NB: effective potential depends on the density, which depends on the potential ...

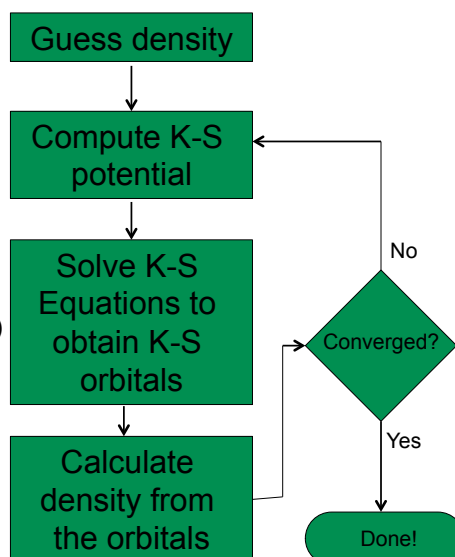
## DFT: Approximations and Theory

- KS-DFT Algorithm:
  - a.k.a. Self-Consistent Field approach (SCF)

$$V_{KS}(\vec{r}) = V_{ext}(\vec{r}) + V_H(n(\vec{r})) + V_{XC}(n(\vec{r}))$$

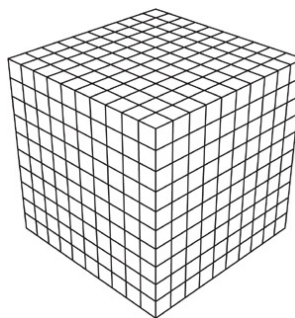
$$\left[ -\frac{1}{2} \nabla^2 + V_{KS}(\vec{r}) \right] \phi_i(\vec{r}) = \epsilon_i \phi_i(\vec{r})$$

$$n(\vec{r}) = \sum_i |\phi_i(\vec{r})|^2$$



## DFT: Implementation

- How do we represent each of the objects in KS-DFT in a computer program?
  - From here-on, assume we are looking at a periodic system (e.g. crystal)
- Electron density is easy – just use a 3D grid
  - Typical grid sizes ~100s of grid points for simulation cell of side 10Å
  - Stored as a 3D array...



## DFT: Implementation

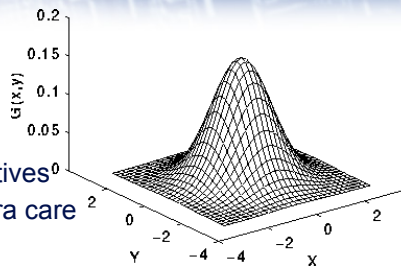
- What about the orbitals ('wave-functions')?
  - We would like:
    - Compact (low memory)
    - Efficient (convenient for numerical algorithms)
    - Accurate
    - Simply discretising on a grid does not give accurate enough derivatives for K.E. term in K-S equations.
  - Generally, write an orbital (or rather the corresponding periodic Bloch function) as a sum of *basis functions*
    - Store just the co-efficients of each basis function
    - Rewrite the K-S equations in matrix form

$$H_j c_{ij} = \epsilon_i c_{ij}$$

## DFT: Implementation

- Choices for basis functions

- Gaussian functions
  - Small number needed per orbital
  - Analytic forms for integrals, derivatives
  - Atom-centred, so need to take extra care
- Numerical basis sets
  - Store a basis function explicitly on a small grid
  - Numerical integration, differentiation (hard to control accuracy)
- Plane Waves
  - Expand each orbital as a Fourier Series, and store G-space coefficients on a regular grid
  - Use FFT to get back to real-space (e.g. to compute density)
  - Large number of basis functions (10,000+) so cannot store H

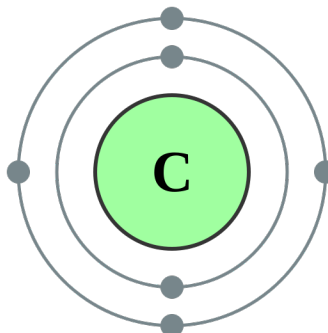


## DFT: Implementation

- Many common plane-wave codes
  - E.g. VASP, CASTEP, Quantum Espresso (PWscf), CPMD, Abinit
- Hamiltonian (total energy) calculated partly in real-space, and partly in Fourier space:
  - Real space:  $E_{XC}$ ,  $E_{II}$
  - Fourier space:  $E_K$ ,  $E_H$ ,  $E_V$
- Cost of calculation  $O(N^2)$ , N the number of electrons
  - For implementations which use a minimization method for solving the K-S Equations
  - Or  $O(N^3)$  if matrix diagonalisation is used

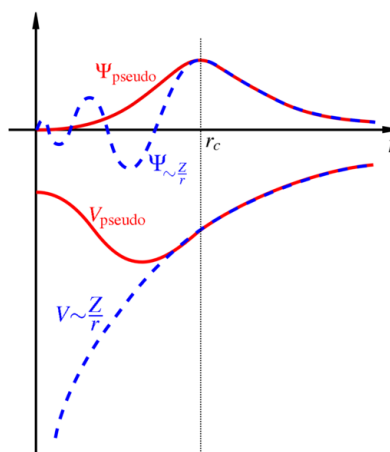
## DFT: Implementation

- Psuedopotentials: reducing N
  - Recall electrons are arranged in 'shells' around the nucleus
  - Small electron density and very large electrostatic potential at the core
  - Would need a very large basis set to represent core electron wave functions
  - Chemistry tells us not to care about them! Model the valence electrons only.



## DFT: Implementation

- Psuedopotentials cont.
  - Frozen core approximation
  - Core electrons and nucleus taken together, reducing electron count and ionic charge
  - Psuedopotential constructed to avoid diverging potential, smoothing the wave-function, but retaining the long-range behaviour.
  - Smooth wave-function requires less PWs to represent.



## Summary

- Comments on Software:

- Vast number of implementation choices has led to a profusion of codes:

70(!) are listed on

[http://en.wikipedia.org/wiki/](http://en.wikipedia.org/wiki/List_of_quantum_chemistry_and_solid-state_physics_software)

[List\\_of\\_quantum\\_chemistry\\_and\\_solid-state\\_physics\\_software](http://en.wikipedia.org/wiki/List_of_quantum_chemistry_and_solid-state_physics_software)

- Many research groups only use one (or a few) codes, many years of effort invested in developing scripts, 'local knowledge', which parameters should be tweaked (and which 'just work')
  - As a result, researchers will not always use the 'best tool for the job'

## Summary

- Comments on Software (cont):

- Benchmarking of applications against each other is hard
  - How to be sure they are solving the 'same' problem?
  - Many code have unique features which make them indispensable (for some problems, for some users)
  - May be specifically forbidden for publishing performance data!
    - <http://www.bannedbygaussian.org/>
- Significant usability problem / steep learning curve for many codes
  - Helped by integration with GUIs, automation tooling, still more to be done...



## Summary

- With care:
  - “First-principles methods may be used for subtle, elegant and accurate computer experiments and insight into the structure and behaviour of matter.”, K. Refson



## Summary

- Without care:
  - “First-principles results may be worthless nonsense”, K. Refson





## Summary

- Materials science / computational chemistry accounts for >50% of the compute time used on HECToR
- Density Functional Theory (and its implementation) are arguably the most significant developments in the field in the last 40 years
- Lots of work (both physical theory and code development) done to produce reasonably efficient codes for modern HPC
- Most codes parallelise (quite) well, but lots for us still to do!

## References

- Payne *et al*, Rev. Mod. Phys. 64, 1045-1097 (1992)
- Richard M. Martin, “*Electronic Structure: Basic Theory and Practical Methods*”, Cambridge University Press, 2004
- K. Refson, “CCP5 Summer School 2011: First Principles Simulation”
- J. Hutter, “*Ab initio Molecular Dynamics and Density Functional Theory*”, 1<sup>st</sup> CP2K CECAM Tutorial, 2009, <http://www.cecam.org/workshop-0-273.html>
- K. Burke, “*The ABC of DFT*”, <http://dft.eci.edu/doc/g1.pdf>