

# Computational Physics

Jan Ryckebusch

Ghent University

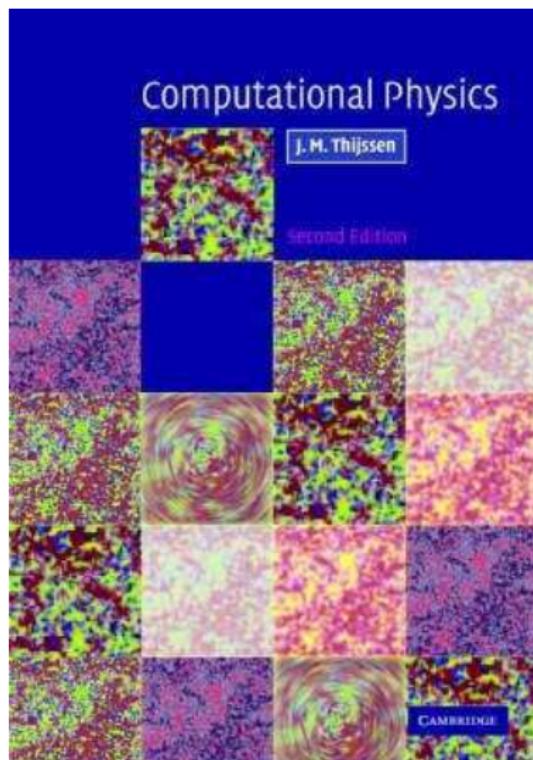
Academic Year 2013-2014



FACULTEIT WETENSCHAPPEN

# COMPUTATIONAL PHYSICS

J.M. Thijssen "Computational Physics"  
(Second Edition, Cambridge University Press, 2007)



- Jos Thijssen, "Delft University of Technology"
- Suitable for graduate students: bridges the gap between undergraduate-level physics and computational research
- Lecturers: Jan Ryckebusch and Toon Verstraelen
- Practicals: Camille Colle, Maarten Vanhalst, Toon Verstraelen

# Computational physics: theory and practicals

- 1 **THEORY CLASS: Week 1 - Week 12**
  - Tuesday, 13.00-14.15 and 14.30-15.45  
(INW, Proeftuinstraat 86)
- 2 **PRACTICAL CLASS: Week 3 - Week 12**
  - ALL STUDENTS: Friday, 14.30-17.15  
(INW, Proeftuinstraat 86)
- 3 **Week 1-2: three theory classes**

# Contents of the course

- Chapter 1: Introduction (p1-13)
- Chapter 2: Quantum scattering with a spherically symmetric potential (p14-29)
- Chapter 3: The variational method for the Schrödinger equation (p29-41)
- Chapter 4: The Hartree-Fock method (p43-88)
- Chapter 5: Density functional theory (p89-121)
- Chapter 7: Classical equilibrium statistical mechanics (p169-196)
- Chapter 8: Molecular dynamics simulations (p197-262)
- Chapter 10: The Monte-Carlo method (p295-337)
- Appendix A: Numerical methods (p557-604)
- Appendix B: Random number generators (p605-613)

# Computational Physics (2013-2014)

WEEK	DATE	TUESDAY	FRIDAY
Week 39	23/09-27/09	TH1: Introduction (JR)	TH2: Scattering 1 (JR)
Week 40	30/09-04/10	TH3: Scattering 2 (JR)	PR1: Linux, Python, GNUpolt (JR)
Week 41	07/10-11/10	TH4: Variational Method (JR)	PR2: Duffing Oscillator (JR)
Week 42	14/10-18/10	PR3: Three-dimensional HO (JR)	TH5: Statistical Mechanics (JR)
Week 43	21/10-25/10	TH6: Molecular Dynamics 1 (JR)	PR4: Scattering 1 (JR)
Week 44	28/10-01/11	TH7: Molecular Dynamics 2 (JR)	HOLIDAY
Week 45	04/11-08/11	TH8: Monte-Carlo 1 (JR)	PR5: Scattering 2 (JR)
Week 46	11/11-15/11	TH9: Monte-Carlo 2 (JR)	HOLIDAY
Week 47	18/11-22/11	PR6: Molecular Dynamics 1 (JR)	PR7: Molecular Dynamics 2 (JR)
Week 48	25/11-29/11	TH10: DFT 1 (TV)	TH11: DFT 2 (TV)
Week 49	03/12-07/12	PR8: DFT 1 (TV)	PR9: DFT 2 (TV)
Week 50	09/12-13/12	---	---

TH: Theory class

PR: Practical class

JR: J. Ryckebusch

TV: T. Verstraelen

# Physics and computational physics

- Very few physical problems have an analytical solution!
- These days, computers are powerful!
- Several physical problems can be solved with numerical techniques!
- **BUT ..., IN MOST CASES ONE NEEDS APPROXIMATIONS TO THE PHYSICAL EQUATIONS !**
- Computational physics: develops approximation methods for physical problems and techniques to implement these approximation methods on a computer!
- In the book: focus on **HOW** these approximations work, **WHY** they work, and **WHAT** the approximations are.
- A extensive list of books on Computational Physics:  
<http://sip.clarku.edu/books/>

# Programming

- Most of the implementations of the algorithms are discussed in a step-by-step fashion making use of metalanguage, and can be programmed in a language of choice
- In the practicals you will most often make use of PYTHON:  
*“Python is a general-purpose, interpreted high-level programming language whose design philosophy emphasizes code readability.”*
- PYTHON is relatively easy to use and compromises between “open” (C++, FORTRAN, JAVA, ...) and “closed” (MATHEMATICA, MATLAB, ...) computer languages
- Scientific visualization software:
  - 1 GNUPLOT: *“gnuplot is a command-line program that can generate two- and three-dimensional plots of functions, data, and data fits. It is frequently used for publication-quality graphics as well as education.”*

# Chapter 1 : Introduction

**“bird’s-eye perspective of different fields of physics and the computational methods used”**

- 1.1 Physics and computational physics
- 1.2 Classical mechanics and statistical mechanics
  - A.7.1 Ordinary differential equations
  - Example 1: Particle subject to friction and gravity
  - Example 2: Driven pendulum
- 1.3 Stochastic simulations
- 1.4 Electrodynamics and hydrodynamics
- 1.5 Quantum mechanics
- 1.6 Relations between quantum mechanics and classical statistical physics
- 1.7 Quantum molecular dynamics
- 1.8 Quantum field theory
- 1.9 About this book

# 1.1 Physics and computational physics

Solving a physical problem often amounts to

- solving a differential equation (*classical mechanics, electrodynamics, quantum mechanics, fluid dynamics, etc.*)
- performing sums of integrals over large numbers of degrees-of-freedom (*statistical physics, field theory, etc.*)
- solving an eigenvalue problem (*quantum mechanics*)

Often, use a numerical routine

For example, from

- Numerical recipes: <http://www.nr.com/>
- Netlib: <http://www.netlib.org/>
- Gnu Scientific Library (GSL) (C library)  
<http://www.gnu.org/software/gsl>

In most cases : approximations appear necessary !

# 1.2 Classical mechanics and statistical mechanics

## Newton's equation of motion

$$m\ddot{x}(t) = F[x(t), \dot{x}(t), t]$$

Initial conditions :  $x(t_0)$ ,  $\dot{x}(t_0)$  and  $\ddot{x}(t_0)$  [can for example be solved with **Verlet Algorithm of Eq. (A.42)**]

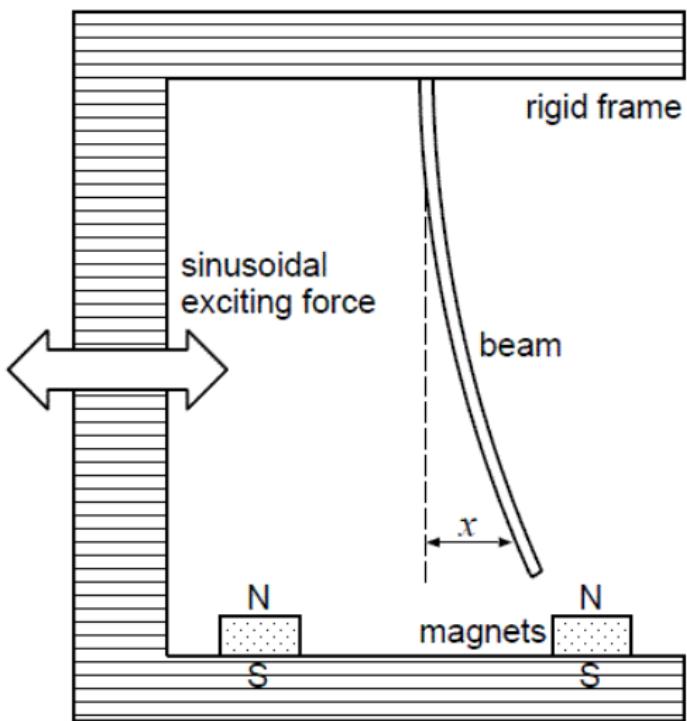
- Chaotic systems

Duffing oscillator :  $F = -\gamma\dot{x} + 2ax - 4bx^3 + F_0 \cos(\omega t)$

- ***N*-body problem:**  $N$  particles subject to each other's interaction
  - 1 gravitational, electrostatic :  $1/r$  potentials
  - 2 analytic solutions for  $N = 2$ .
- **Molecular dynamics:** Many interacting atoms and molecules (gases, liquids, solids). Results : equation-of-state  $P(V, T, N)$ , pair correlation functions  $g(r_1, r_2)$  (Chapter 8).

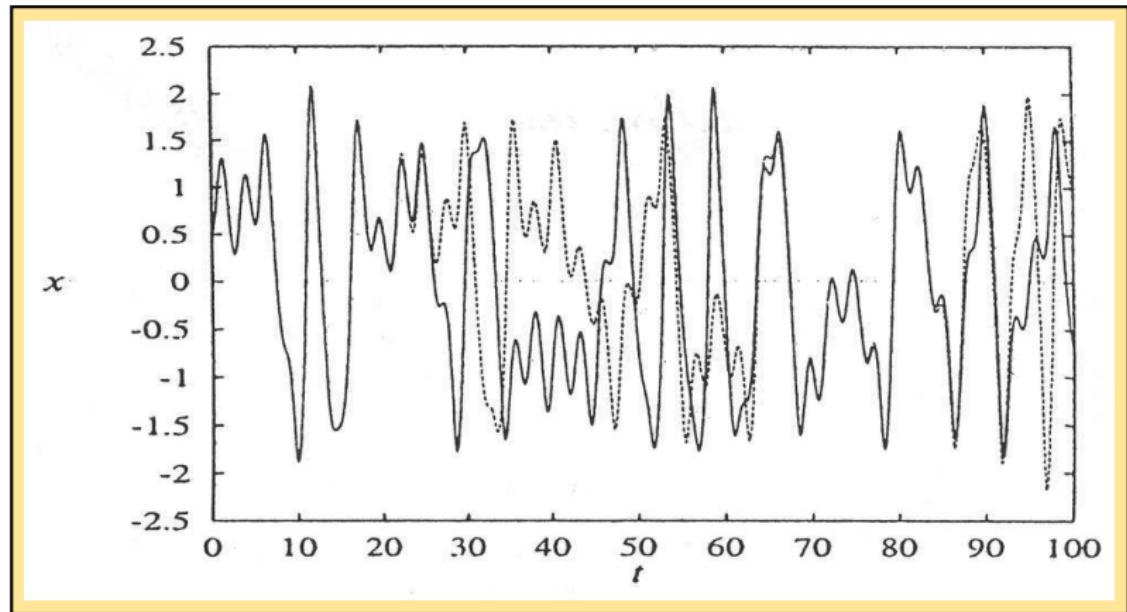
# Duffing oscillator (1): experimental realization

- Particle subject to a periodic force + friction + Mexican-hat potential  $V(x) = bx^4 - ax^2$
- For  $a, b > 0$ : The Duffing oscillator can be regarded as a model of a periodically forced steel beam deflected toward two magnets.
  - gravity + magnets: double potential well
  - “air” provides the frictional force



## Duffing oscillator (2): Chaotic behaviour

Duffing oscillator exhibits chaotic behaviour for particular values of  $\gamma, a, b, F_0, \omega$  !



Motion is irregular and very small changes in the initial conditions have dramatic consequences!

## A.7 Differential equations (p568-590)

- Linear differential equations (DE):  $\mathcal{L}\psi = F$ 
  - 1  $F$ : known function
  - 2  $\mathcal{L}$ : linear combination of derivatives
  - 3  $\psi$ : unknown function or solution
- Two types of differential equations
  - 1 ODE: Ordinary Differential Equations (*one variable*)
  - 2 PDE: Partial Differential Equations (*more variables*)
- Law of linear superposition for linear DE: solutions can be added together to form other solutions

Differential equations of second order occur most frequently in physics (acceleration in classical mechanics, kinetic energy in quantum mechanics, Maxwell's (Dirac's) equations after eliminating one of the unknown functions, ...).

## A.7 Differential equations (p568-590)

### What makes a method to solve a DE a good one?

(depends on the problem at hand ...)

- 1 **Precision and speed:** high precision will cost more time ; higher speed will yield less precise results
- 2 **Stability:** method is unstable if errors tend to grow during integration (source of errors: starting values and discrete numerical representation)
- 3 **Implementation:** complicated algorithms may consume a lot of time to implement
- 4 **Flexibility:** some methods demand a fixed discretisation interval
- 5 **Symmetry:** numerical method to share symmetry properties of the original equation  
*(symmetry properties are related to conservation laws!)*

## A.7.1. Ordinary differential equations (p569)

- First-order ODE

$$\dot{x}(t) = f[x(t), t]$$

- Second-order ODE

$$\ddot{x}(t) = f[x(t), t]$$

can be transformed into a set of two differential equations of the first order

$$\begin{cases} \dot{x}(t) = y(t) \\ \dot{y}(t) = f[x(t), t] \end{cases}$$

## A.7.1 Stability of ODE (p568)

- 1 first order, one dimension: small time step guarantees stability!!
- 2 higher orders and/or dimensions: stability is an issue!!!
  - equations have more than one independent solution
  - initial conditions determine how these are combined in the actual solution
  - e.g. the solution we are after is a damped one, but ... growing solution mixes in and spoils things (can often be solved by integrating in the other direction)
  - problems can be severe when two solutions have a very different time scale! (e.g. combination of slow and fast solutions)
  - very strongly damped term in the solution may be easily overlooked ("stiff equations")
  - ...

## A.7.1 The Runge-Kutta method (I)

We look for solutions of an ODE of the type  $\dot{x}(t) = f[x(t), t]$

- Euler's forward method

$$x(t = h) = x(t = 0) + hf[x(t = 0), t = 0] + \mathcal{O}(h^2)$$

- Runge-Kutta method can be considered as a step-wise improvement of Euler's forward method
- Midpoint method [Wiki page](#)

$$\begin{aligned}k_1 &= hf[x(0), 0] = x(h) - x(0) \\k_2 &= hf \left[ x(0) + \frac{x(h) - x(0)}{2}, \frac{h}{2} \right] \\&= hf \left[ x(0) + \frac{1}{2}k_1, \frac{h}{2} \right] \\x(h) &= x(0) + k_2 + \mathcal{O}(h^3)\end{aligned}$$

Error is of the order  $h^3$ : can be verified by expanding  $x(t)$  in a power series in  $t$  to second order.

## A.7.1 The Runge-Kutta method (II)

Error analysis of the midpoint method for  $\dot{x}(t) = f[x(t), t]$

- One has:  $x(h) = x(0) + \int_0^h dt \dot{x}(t)$
- Taylor expansion of the function  $\dot{x}(t)$  about  $t = \frac{h}{2}$

$$x(h) \approx x(0) + \int_0^h dt \left[ \dot{x}\left(\frac{h}{2}\right) + \left(t - \frac{h}{2}\right) \ddot{x}\left(\frac{h}{2}\right) + \frac{\left(t - \frac{h}{2}\right)^2}{2} \ddot{x}\left(\frac{h}{2}\right) \right]$$

- Second term in the integral vanishes, and one has

$$x(h) = x(0) + hf \left[ x\left(\frac{h}{2}\right), \frac{h}{2} \right] + \theta(h^3)$$

- As  $x\left(\frac{h}{2}\right)$  is not known

$$x(h) = x(0) + hf \left[ x(0) + \frac{h}{2} f[x(0), 0], \frac{h}{2} \right] + \theta(h^3)$$

## A.7.1 The Runge-Kutta method (II)

- advantage of Runge-Kutta method: at each integration step one can use different time interval  $h$ !
- disadvantages of Runge-Kutta method: SLOW because function  $f[x(t), t]$  has to be evaluated several times in each evaluation step

## A.7.1 (p572) Verlet algorithm (I)

- Verlet algorithm is a simple and efficient method for integrating second-order differential equations of the form

$$\ddot{x}(t) = F[x(t), t]$$

- Taylor expansion:

$$x(\pm h) = x(0) \pm h \dot{x}(0) + \frac{h^2}{2} F[x(0), 0] \pm \frac{h^3}{6} \ddot{x}(0) + O(h^4)$$

$O(h^4)$ : deviation of the exact solution from the numerical one is smaller than  $\alpha_n h^4$  with  $\alpha_n$  finite numbers which are nonvanishing for  $h \rightarrow 0$ .

- One finds:

$$x(h) = 2x(0) - x(-h) + h^2 F[x(0), 0] + O(h^4)$$

**Knowing two positions ( $x(0)$  and  $x(-h)$ ) one can determine  $x(h)$ .**

## A.7.1 (p575) Verlet algorithm (II)

- It can be shown that the accumulated error in the position after a large number of integration steps is of the order  $O(h^2)$ .
- Velocity can be determined from (good enough!)

$$\dot{x}(0) = v(0) = \frac{x(h) - x(-h)}{2h} + O(h^2)$$

- Accordingly:

$$x(h) = x(0) + hv(0) + \frac{h^2}{2} F[x(0), 0] + O(h^3)$$

**Knowing one position ( $x(0)$ ) and one velocity ( $v(0)$ ) one can determine  $x(h)$ .**

## A.7.1 (p575) Verlet algorithm (III)

- Verlet algorithm is invariant under time reversal
  - 1 under the substitution  $[x(0), x(h) \Leftrightarrow x(h), x(0)]$  exactly the same path in phase space will be followed but backward in time
  - 2 in practice, small deviations as a result of finite numerical precision!
- Disadvantage of Verlet algorithm: time step  $h$  is not easily adapted in the Verlet algorithm! (possible through rescaling of the time  $t \rightarrow t' = \alpha t$ , or using interpolation to construct new starting values in between two known values)
- For oscillatory solutions: time step  $h$  has to be taken sufficiently small to avoid instabilities!

# Example I: Particle subject to gravity and friction (I)

- Equation to solve ( $v \equiv |\dot{\vec{r}}|$ ):

$$m\ddot{\vec{r}} = -mg\vec{e}_z - bv^n \frac{\dot{\vec{r}}}{v}$$

- Friction in the opposite direction of  $\vec{v} = \dot{\vec{r}}$
- $v^n$ :  $n \approx 1$  for “low”  $v$ ,  $n \approx \frac{3}{2}$  for “medium”  $v$ ,  $n \approx 2$  for “high”  $v$ ,
- Initial conditions are initial positions and initial velocities:  
 $(x(t_0), \dot{x}(t_0), z(t_0), \dot{z}(t_0))$  in the two-dimensional case
- In the absence of friction the problem can be solved analytically

$$\begin{cases} x(t) = x(t_0) + \dot{x}(t_0)(t - t_0) \\ z(t) = z(t_0) + \dot{z}(t_0)(t - t_0) - \frac{g(t-t_0)^2}{2} \end{cases}$$

# Example I: Particle subject to gravity and friction (II)

- Set of differential equations (in 2D):

$$\begin{cases} \ddot{x}(t) &= f(x(t), \dot{x}(t), z(t), \dot{z}(t)) \\ &= -\frac{b}{m} \left( \sqrt{\dot{x}(t)^2 + \dot{z}(t)^2} \right)^{n-1} \dot{x} \\ \ddot{z}(t) &= g(x(t), \dot{x}(t), z(t), \dot{z}(t)) \\ &= -\frac{b}{m} \left( \sqrt{\dot{x}(t)^2 + \dot{z}(t)^2} \right)^{n-1} \dot{z} - g \end{cases}$$

Initial conditions:  $(x(t_0), \dot{x}(t_0), z(t_0), \dot{z}(t_0))$

- Step 1 of solution strategy: turn the system of second-order ODEs into a system of first-order ODEs

$$\begin{cases} (i) \quad \dot{x}(t) &= v_x(t) \\ (ii) \quad \dot{v}_x(t) &= f(x(t), v_x(t), z(t), v_z(t)) \\ (iii) \quad \dot{z}(t) &= v_z(t) \\ (iv) \quad \dot{v}_z(t) &= g(x(t), v_x(t), z(t), v_z(t)) \end{cases}$$

# Example I: Particle subject to gravity and friction (III)

- Simple numerical method to solve the set of ODE's: EULER method (often fine for small  $dt$ , error of the order  $\theta(dt^2)$ )

$$\begin{cases} x(t_0 + dt) = x(t_0) + dt \times v_x(t_0) \\ v_x(t_0 + dt) = v_x(t_0) + dt \times f(x(t_0), v_x(t_0), z(t_0), v_z(t_0)) \\ z(t_0 + dt) = z(t_0) + dt \times v_z(t_0) \\ v_z(t_0 + dt) = v_z(t_0) + dt \times g(x(t_0), v_x(t_0), z(t_0), v_z(t_0)) \end{cases}$$

- Given the initial conditions:  $(x(t_0), v_x(t_0), z(t_0), v_z(t_0))$
- One can solve the problem for a sequence of time points between  $t_0$  and  $t_f$ :  $t_0 \rightarrow t_0 + dt \rightarrow t_0 + 2dt \rightarrow \dots \rightarrow t_f$
- Practical implementation I: MOTIONOPARTICLE.PY (example of the use of Python in an open environment: the numerical method, the integrator for the set of ODEs, is developed by the user)

## Example I: Particle subject to gravity and friction (IV)

- Practical implementation II: MOTIONOFPARTICLECLOSED.PY  
(example of the use of Python in a closed environment: the numerical method is provided by the software)
- The function CALL ODEINT(DERIV, ZINIT, TIME) handles all the integrations.
- ODEINT(FUNC, Y0, T): Integrates a system of first order ODEs

$$\frac{d\vec{y}}{dt} = \text{func}(\vec{y}, t)$$

- 1 FUNC(Y, T0) computes the derivative of y at t0.
- 2 Y0: array with initial condition on y (can be a vector).
- 3 T: array (A sequence of time points for which to solve for y. The initial value point should be the first element of this sequence.)
- It is possible to tell ODEINT which integrators to use.

# Example I: Particle subject to gravity and friction (V)

- Equation to solve ( $v \equiv |\dot{\vec{r}}|$ ):

$$m\ddot{\vec{r}} = -mg\vec{e}_z - bv^2 \frac{\dot{\vec{r}}}{v}$$

- The  $\vec{y}$  in ODEINT

$$\vec{y} = [y[0], y[1], y[2], y[3]] = [x, v_x, z, v_z]$$

- The set of first-order ODEs

$$\begin{aligned}\frac{d\vec{y}}{dt} &= \left[ v_x, -\frac{b}{m} \sqrt{v_x^2 + v_z^2} v_x, v_z, -\frac{b}{m} \sqrt{v_x^2 + v_z^2} v_z - g \right] \\ &= \left[ y[1], -\frac{b}{m} \sqrt{y[1]^2 + y[3]^2} y[1], \right. \\ &\quad \left. y[3], -\frac{b}{m} \sqrt{y[1]^2 + y[3]^2} y[3] - g \right]\end{aligned}$$

## Example II: Driven pendulum (I)

- The driven pendulum is another example of a system that exhibits chaotic behaviour

$$\ddot{\varphi} = \sin \varphi + \frac{1}{4} \cos t$$

- Solution to this equation:  $\varphi(t)$
- Second-order ODE can be written as a set of two first-order ODEs

$$\begin{cases} \dot{\varphi} = v \\ \dot{v} = \sin \varphi + \frac{1}{4} \cos t \end{cases}$$

- We use an ODE solver from the SCIPY module of Python to solve this set of coupled equations and plot the results
- Detailed information about SCIPY: <http://www.scipy.org/>

# Example II: Driven pendulum (II)

## The Python routine for the Driven pendulum (part 1)

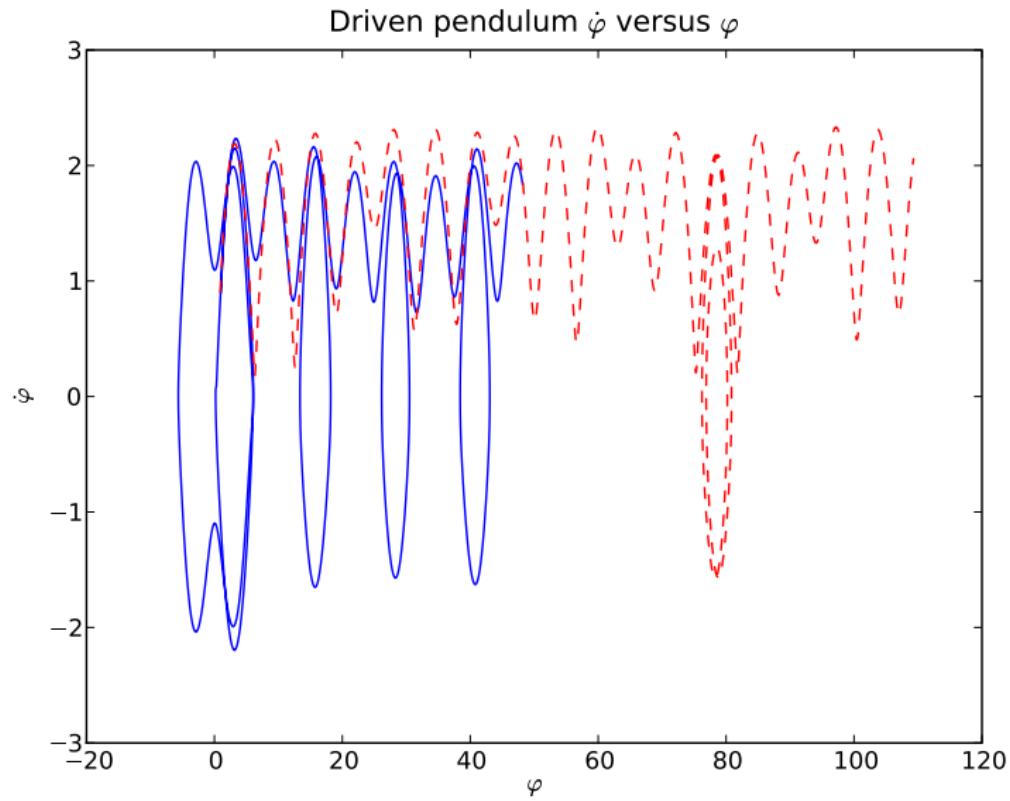
```
#  
# Python routine for the driven pendulum  
#      solves \ddot{\phi} = \sin \phi + \frac{1}{4} \cos t  
#  
#  
from pylab import *          # import the plotting routines  
from scipy.integrate import odeint # import the ODE solver from scipy  
#  
# lines starting with a # are comments  
#  
#  
#  
# define a function that returns the derivative  
#  
def derivative(y, t):  
    return array([y[1], sin(y[0]) + 0.25 * cos(t)])  
#  
#  
# define a function that solves the ODE for the forced pendulum  
#  
def compute_trajectory(y0):  
    t = arange(0.0, 100.0, 0.1) # an array from 0 to 100 in steps of 0.1  
    y_t = odeint(derivative, y0, t)  
    return y_t[:, 0], y_t[:, 1] #[:,0] full first column of a list
```

# Example II: Driven pendulum (III)

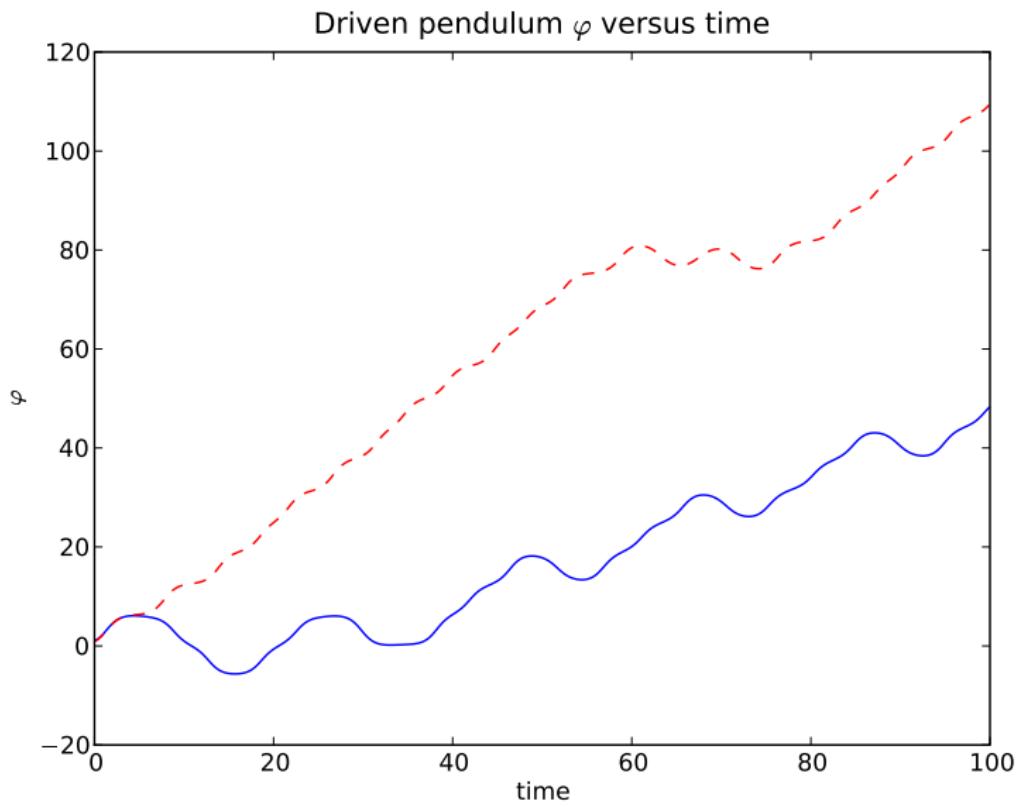
## The Python routine for the Driven pendulum (part 2)

```
#  
#  
# the main program: solves the ODE and makes a plot  
#  
phi_a, v_a = compute_trajectory([1.0, 0.9])  
phi_b, v_b = compute_trajectory([0.9, 0.9])  
#  
# start producing the plots  
# (phi versus v)  
plot(phi_a, v_a)  
plot(phi_b, v_b, "r--") # v_b as a function of phi_b with a red dashed line  
xlabel(r"\varphi")  
ylabel(r"\dot{\varphi}")  
title(r'Driven pendulum $\dot{\varphi}$ versus $\varphi$')  
show()  
#  
# (phi versus time)  
#  
tt = arange(0.0, 100.0, 0.1)  
plot(tt, phi_a)  
plot(tt, phi_b, "r--") # v_b as a function of phi_b with a red dashed line  
xlabel("time")  
ylabel(r"\varphi")  
title(r'Driven pendulum $\varphi$ versus time')  
show()
```

## Example II: Driven pendulum (IV)



## Example II: Driven pendulum (V)



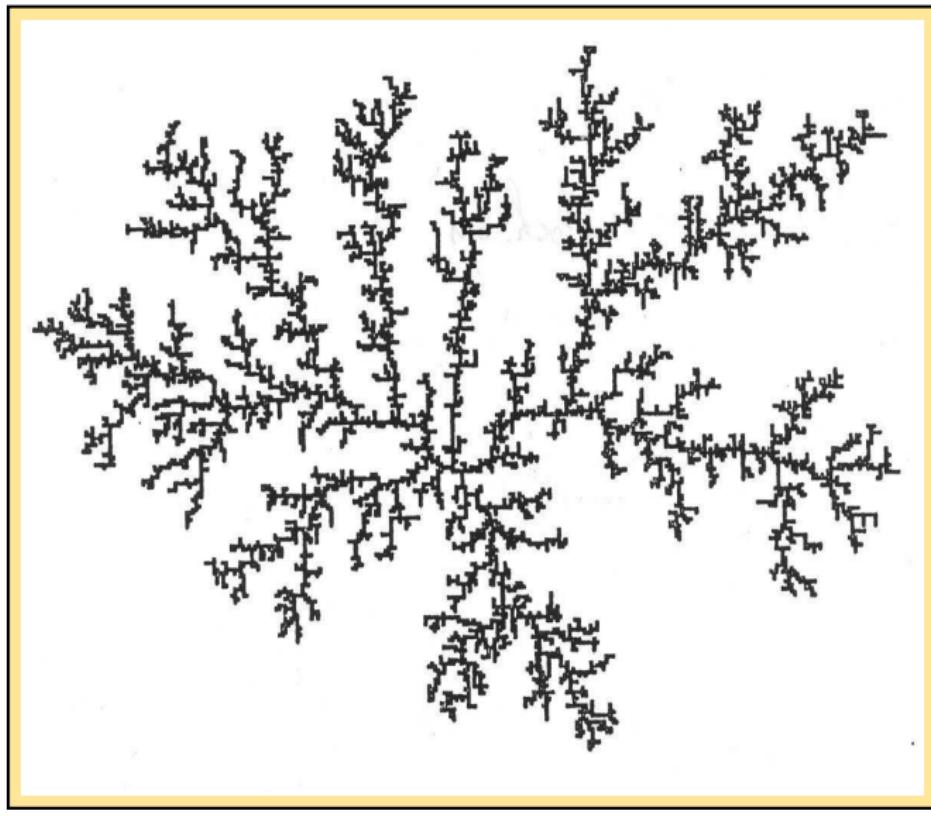
# 1.3 Stochastic simulations

- **Monte Carlo method** : procedures in which random numbers play an essential role. For example : move particles around in a random fashion if interactions are not precisely known (e.g. in some liquids).

Remark: MC uses artificial dynamics and is not suitable for studying dynamic behaviour!

- Diffusion limited aggregation (DLA): the use of random number generators for simulations. Simulation of a dendritic cluster (crystals, bacterial colonies, frost patterns on windows, ...).  
Simulation : <http://apricot.polyu.edu.hk/dla/dla.html>
- Monte-Carlo method used for studying
  - 1 many-particle systems at a predefined temperature, volume and particle number
  - 2 lattice spin models (magnetism and field theory)

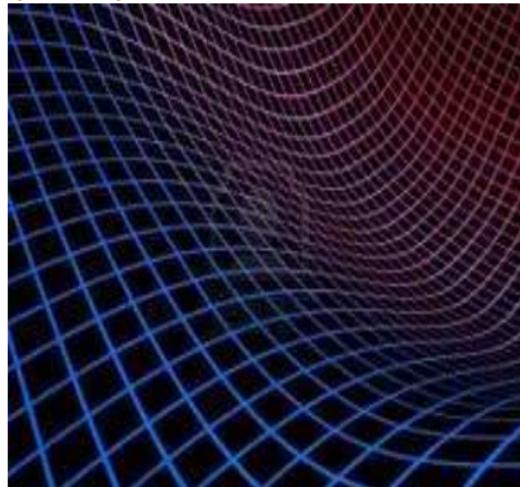
# Diffusion Limited Aggregation



Interesting physics can be studied by applying simple algorithms!

# 1.4 Electrodynamics and Hydrodynamics

EQUATIONS ARE PARTIAL  
DIFFERENTIAL EQUATIONS  
(PDE)!



Engineering Problems

numerical methods for solving PDEs are abundant

- Fields are **continuous** and an infinite amount of variables is involved!
  - 1 Discretization methods (solve on a grid)
  - 2 Alternatively, expand in terms of smooth functions and determine the finite number of expansion coefficients
- Partial differential equations
  - 1 Finite Difference Methods (FDM)
  - 2 Finite Element Methods (FEM)
  - 3 Fourier Transform Methods (FTM)
  - 4 Multigrid Methods
  - 5 ...

# 1.5 Quantum mechanics (I)

Solve the Schrödinger equation for one or more particles (external potential felt by the particles, and/or interactions amongst the particles)

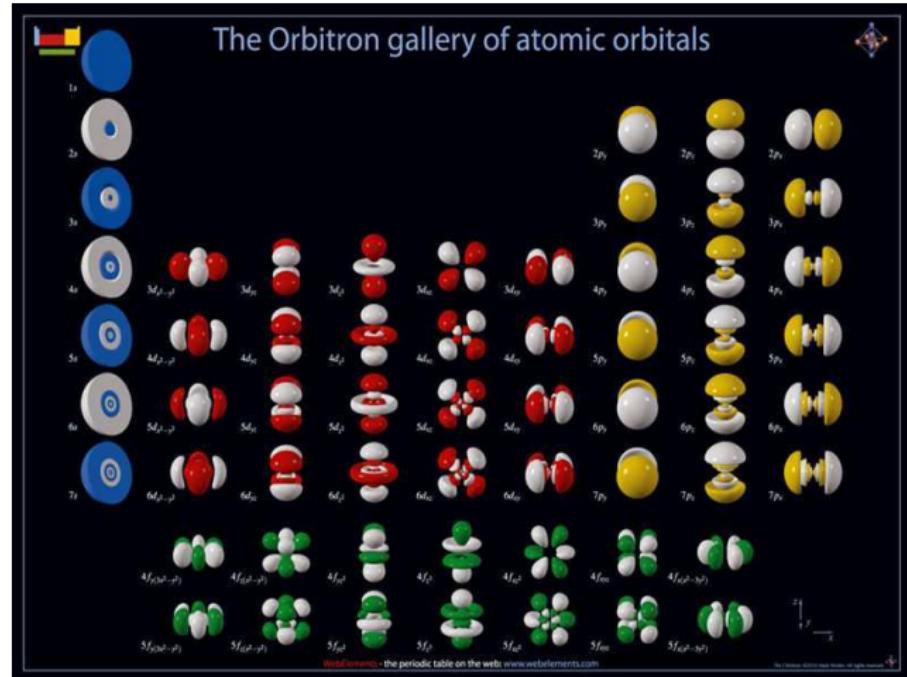
- One particle, one dimension: ordinary differential equation.
  - 1 Techniques similar to those used in solving Newton's equations
  - 2 But ... stationary Schrödinger equation is an eigenvalue problem: the energy  $E$  must be varied until wave functions matches some boundary conditions and is normalisable
- More than one particle, more than one dimension : partial differential equations
  - 1 Sometimes, problem can be reduced to ordinary differential equation
  - 2 Example: Scattering in a spherically symmetric potential (Chapter 2)

# 1.5 Quantum mechanics (II)

**Behaviour of many electrons moving in the field generated by nuclei. Discretization methods with grids which are finer close to the nucleus where the wave functions vary strongly.**

## Variational method

expand the wave function in terms of a proper set basis functions and find expansion coefficients.  
(Chapter 3)



# 1.5 Quantum mechanics (III)

***Behaviour of many electrons interacting via the Coulomb potential and moving in the field generated by nuclei.***

- Hartree-Fock (HF) : antisymmetry OK, independent-particle model, **poor treatment of correlations !**
- Density-functional theory (DFT) : **antisymmetry  $\approx$ , unphysical electron self-interaction**, correlations OK
- Both HF and DFT amount to solving **MATRIX EIGENVALUE PROBLEM**
- Solutions: energy eigenvalues (ground state and excited states) AND corresponding wave functions
- energy eigenvalues can be directly compared with experiment
- DFT and HF are also heavily used in Nuclear Physics and Molecular Physics

# 1.6 Relations between quantum mechanics and classical statistical mechanics

## STATISTICAL MECHANICS

### Classical

$$Z_{Cl}(T, V, N) = \sum_X \exp -\frac{\mathcal{H}(X)}{k_B T}$$

### Quantum system

$$Z_{QM}(T, V, N) = \text{Tr} \left( \exp -\frac{\hat{H}}{k_B T} \right)$$

Path-integral formalism :  $Z_{QM}$  transformed in  $Z_{Cl}$ -like format !

QUANTUM TIME-EVOLUTION OPERATOR :  $\hat{U}(t, t_0 = 0) = \exp -\frac{i}{\hbar} \hat{H}t$

Analogy with partition function :

$$\frac{it}{\hbar} \leftrightarrow \frac{1}{k_B T}$$

# Wick rotations (I)

- 1 Problem in Minkowski space can be translated to a problem in Euclidean space (often more simple) by means of replacing  $t$  by  $i\tau$  (imaginary time)

$$ds^2 = dx^2 + dy^2 + dz^2 - dt^2 \quad \text{Minkowski metric}$$

$$ds^2 = dx^2 + dy^2 + dz^2 + d\tau^2 \quad \text{Euclidean metric}$$

- 2 Problem in statistical mechanics can be connected to quantum mechanics by replacing  $\beta = \frac{1}{k_B T}$  with imaginary time  $\frac{it}{\hbar}$

- Average value of some quantity  $Q$  in statistical mechanics

$$\langle Q \rangle = \frac{\sum_j Q_j \exp -\beta E_j}{\sum_j \exp -\beta E_j}$$

- Quantum mechanics: Say that at  $t = 0$  the system is created in a uniform superposition state  $|\Psi(t=0)\rangle = \sum_j |j\rangle$ , probability that at  $t$  state is in a state  $|Q(t)\rangle = \sum_j Q_j |j\rangle$

$$\langle \Psi(t) | Q(t) \rangle = \sum_j Q_j \exp -\frac{i}{\hbar} E_j t$$

# Wick rotations (II)

- 3 Wick rotation relates statics' problems in  $n$  dimensions to dynamics' problems in  $(n - 1)$  dimensions by means of replacing one dimension of space with imaginary time!  
( $x \rightarrow it$ )
- 4 Principle of “least energy” in statics becomes the principle of “least action” in dynamics

Wiki page on Wick rotations!



## 1.7 Quantum molecular dynamics

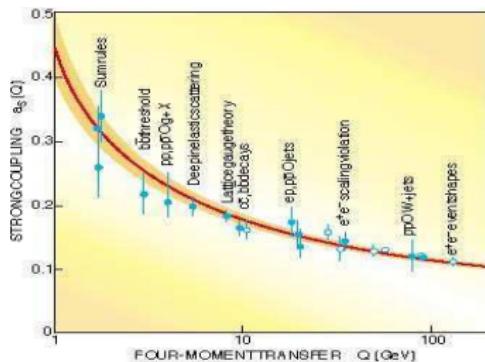
- many interacting atoms or molecules can be studied classically by solving Newton's equations of motion
- these simulations adopt “pair interactions” to describe atomic interactions
- pair potentials can be determined through quantum mechanical calculations (HF, DFT, QMC)
- in dense systems: pair interactions are inadequate as they are influenced by other particles
- forces from full electronic structure calculations including all possible configurations
- Car and Parrinello: method based on MD and DFT to compute these forces approximately as the simulation proceeds (local density approximation)

## 1.8 Quantum field theory (QFT) (I)

- Quantum description of fields: strings in one dimension, sheets in two, ...
- QFT is believed to describe elementary particles and their interactions
- Quantum electrodynamics (QED) works well since perturbation theory can be used
- Quantum chromodynamics (QCD, or theory of the strong interaction): the coupling constant grows for large scales (low energies) and perturbation theory breaks down
- QCD can be “solved” on a lattice: discretize space and time and simulate the theory on a computer (*requires enormous amounts of hardware and often dedicated computing infrastructure*)

# Running coupling constant ...

As momentum transfer decreases  
(probing larger distance scales)  
coupling constant grows



$$\mathcal{L} = \frac{1}{4g^2} G_{\mu\nu}^a G_{\mu\nu}^a + \sum_j \bar{q}_j (i \gamma^\mu D_\mu + m_j) q_j$$

where  $G_{\mu\nu}^a \equiv \partial_\mu A_\nu^a - \partial_\nu A_\mu^a + f_{abc}^a A_\mu^b A_\nu^c$

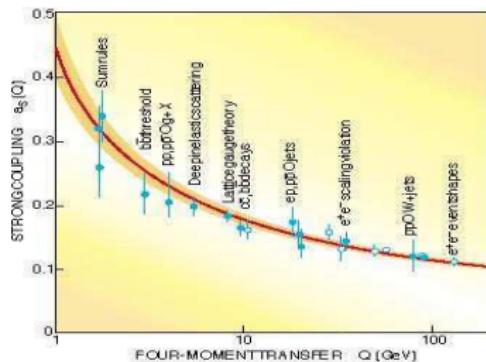
and  $D_\mu \equiv \partial_\mu + i t^a A_\mu^a$

That's it!

- $q_j, m_j$ : quantum field and mass of the quark of  $j$ th flavor
- $A$  is the gluon field with space-time indices and  $a, b, c$  color indices
- $g, m_j$  are the only parameters of the theory!
- numerical coefficients  $f$  and  $t$  guarantee SU(3) color symmetry

# Running coupling constant ...

As momentum transfer decreases  
(probing larger distance scales)  
coupling constant grows



Statement “Higgs boson gives us mass” is incorrect

- 1 Higgs field (not particle) gives mass
- 2 most of atomic mass comes from quarks binding together

$$\mathcal{L} = \frac{1}{4g^2} G_{\mu\nu}^a G_{\mu\nu}^a + \sum_j \bar{q}_j (i \gamma^\mu \partial_\mu + m_j) q_j$$

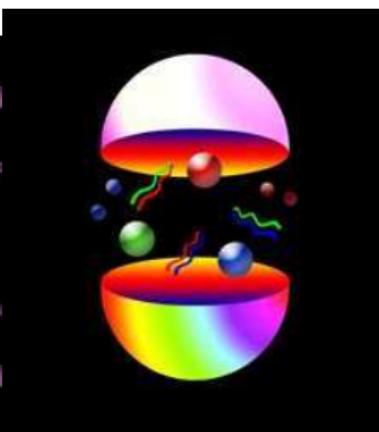
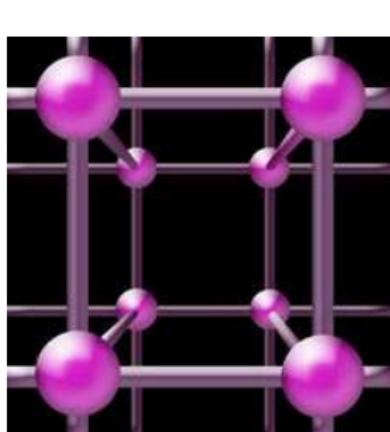
where  $G_{\mu\nu}^a \equiv \partial_\mu A_\nu^a - \partial_\nu A_\mu^a + f_{bc}^{(a)} A_\mu^b A_\nu^c$

and  $D_\mu = \partial_\mu + i t^a A_\mu^a$

That's it!

- $q_j, m_j$ : quantum field and mass of the quark of  $j$ th flavor
- $A$  is the gluon field with space-time indices and  $a, b, c$  color indices
- $g, m_j$  are the only parameters of the theory!
- numerical coefficients  $f$  and  $t$  guarantee SU(3) color symmetry

# 1.8 Quantum field theory (QFT) (II)



QCD simulations relate (bare) quark masses to masses and interaction constants of hadrons!

# Chapter 2 : Quantum scattering with a spherically symmetric potential

- 2.0 “Scattering theory” (Chapter 12 from Robert Scherrer “Quantum Mechanics: an accessible introduction”)
- 2.1 Introduction
- 2.2 A program for calculating cross sections  
(example: *Hydrogen-Krypton scattering*)
  - 2.2.1 Numerov’s algorithm for the radial Schrödinger equation
  - 2.2.2 The spherical Bessel functions
  - 2.2.3 Putting the pieces together
- 2.3 Calculation of scattering cross sections  
(*analytical derivations*)

# Computational physics and quantum scattering

Quantum scattering is a nice representative example of a subject which is addressed in Computational Physics

- Physics consists of three major branches: experimental, theoretical and computational
- Computational physics has the flavors of both experimental and theoretical physics
  - good theoretical background is needed to study a subject by computational methods
  - good computational method comes from a thorough theoretical analysis
  - analysis of the results from numerical calculations not very different from analyzing experimental data (requires a sound theoretical background)

## 2.0 Scattering theory

- Scattering: behaviour of a particle with energy  $E$  incident on a fixed potential  $V$
- In most cases the three-dimensional Schrödinger equation cannot be solved exactly
- Approximation methods
  - 1 Born approximation:  $E \gg V$
  - 2 partial waves: any  $E$  but most valuable for low-energy scattering ( $E < V$ )

### Chapter 12 of Scherrer: “Scattering theory”

12.1 Definition of a cross section

12.2 The Born approximation

12.3 Partial waves

## 12.1 Definition of a cross section (I)

### Some concepts of scattering from classical mechanics

- a region of space is filled with targets (number density  $n$ ) and travels a distance  $L$
- a particle enters this region
- probability that the particle will strike one of the targets

$$P = nL\sigma$$

$\sigma$ : total cross section for scattering

- $\sigma$  is the area of the target that the projectile sees head-on
- $N_i$  incident particles and  $N_s$  scattered ones

$$N_s = N_i n L \sigma$$

- directional information:  $dN_s$  number of particles scattered in  $d\Omega(\theta, \phi)$

$$dN_s(\theta, \phi) = N_i n L \frac{d\sigma}{d\Omega} d\Omega$$

$\frac{d\sigma}{d\Omega}$ : differential cross section (probability that the scattered particle will be scattered in the  $(\theta, \phi)$  direction!)

## 12.2 The Born approximation (I)

- scattering: incident particle with energy  $E$  scattering off a (static) potential  $V(\vec{r})$
- in practice Schrödinger equation is difficult to solve
- Born approximation: variant of perturbation theory that is applicable for  $E \gg V(\vec{r})$  and  $\lim_{r \rightarrow 0} V(r) \approx 0$
- scattering process has time dependence
  - 1 initially: incident particle sees  $V = 0$
  - 2 at time  $t = 0$ : incident particle sees region  $V \neq 0$
  - 3 at large times: incident particle sees  $V = 0$
- Probability that the particle has scattered from  $\psi_i$  to  $\psi_f$  at time  $t$  (time-dependent perturbation theory)

$$P(i \rightarrow f) = \frac{1}{\hbar^2} \left| \int_0^t dt' \langle \psi_f | V(\vec{r}) | \psi_i \rangle \exp \frac{i}{\hbar} (E_f - E_i) t' \right|^2$$

Potential  $V(\vec{r})$  is treated as a small perturbation!

## 12.2 The Born approximation (II)

Several further steps before result can be turned into a differential cross section

- Scattering wave functions (plane waves) must be normalized

$$\int_V \psi_i^* \psi_i d\vec{r} = 1 \quad \psi_i = \frac{1}{\sqrt{V}} \exp i\vec{k}_i \cdot \vec{r}$$

- The incident particle energy can be controlled  $E_i = \frac{\hbar^2 k_i^2}{2m}$ , but not the final energy

$$N(E)dE = \frac{V}{4\pi^2} \frac{(2m)^{3/2}}{\hbar^3} \sqrt{E} dE$$

$N(E)dE$ : number of scattering states in  $[E, E + dE]$

- density of final states (phase space) is proportional to  $\frac{d\Omega_f}{4\pi}$

## 12.2 The Born approximation (III)

Transition probability  $dP$  for scattering in  $d\Omega_f$

$$dP(i \rightarrow \text{any } E_f) = \int P(i \rightarrow f) N(E_f) dE_f \frac{d\Omega_f}{4\pi}$$

- the integration over  $dE_f$  can be worked out with

$$\lim_{a \rightarrow \infty} \frac{\sin^2 ax}{x^2} = \pi a \delta(x)$$

this means that  $E_f = E_i$  which is compatible with a weak perturbation

- from  $P = nL\sigma$  one gets  $dP = n v_i t \frac{d\sigma}{d\Omega_f} d\Omega_f$

## 12.2 The Born approximation (IV)

### Differential cross section in the Born approximation

$$\frac{d\sigma}{d\Omega} = \left( \frac{m}{2\pi\hbar^2} \right)^2 \left| \int d\vec{r} V(\vec{r}) \exp i(\vec{k}_i - \vec{k}_f) \cdot \vec{r} \right|^2$$

For the momentum transfer  $\vec{K} = \vec{k}_f - \vec{k}_i$  one has  $K = 2k \sin \frac{\theta}{2}$  (as  $k_i = k_f = k$ )

- For a spherically symmetric potential one has

$$\frac{d\sigma}{d\Omega} = \frac{4m^2}{\hbar^2 K^2} \left( \int_{r=0}^{r=\infty} dr r V(r) \sin(Kr) \right)^2$$

## 12.3 Partial waves (I)

- scattering: incident particle with energy  $E$  scattering off a (static) spherical symmetric potential  $V(r)$  centered at the origin ( $\lim_{r \rightarrow \infty} V(r) = 0$ )
- in practice Schrödinger equation is difficult to solve
- approximation method: expansion in partial waves (highly effective at low energies, but applicable at any energy)
- incident particle moves with momentum  $k$  along the positive  $z$  axis
- what about the wave function  $\psi_f$  for the scattered particle?

## 12.3 Partial waves (II)

- Idealized situation:  $\psi_f$  is fully spherically symmetric which implies that  $l = 0$  and the radial Schrödinger equation becomes  $\left(E = \frac{\hbar^2 k^2}{2m}\right)$  at large values of  $r$  (or,  $V(r) = 0$ )

$$\frac{d^2}{dr^2} (rR(r)) + k^2 (rR(r)) = 0$$

- 1 SOLUTION:  $R(r) = A \frac{e^{ikr}}{r} + B \frac{e^{-ikr}}{r}$   
Wave expanding with equal probability in all directions!
- 2 the  $B = 0$  component is NOT physical (there are no sources at  $r = \infty$ )

  - In reality: the  $\psi_f$  need no longer be isotropic and we have far from the origin

$$\psi_f = f(\theta) \frac{e^{ikr}}{r}$$

the  $f(\theta)$  does not depend on  $\phi$  as we have assumed a spherically symmetric potential

## 12.3 Partial waves (III)

- far from the origin we have

$$\psi_T = \psi_i + \psi_f = e^{ikz} + f(\theta) \frac{e^{ikr}}{r}$$

- the cross section can be expressed in terms of the probability current densities

$$\frac{d\sigma}{d\Omega} = \frac{|\vec{J}_f|}{\text{solid angle}} = \frac{r^2 |\psi_f|^2}{|\psi_i|^2} = |f(\theta)|^2$$

Probability current density:

$$\vec{J} = \frac{\hbar}{2mi} [\psi^* (\nabla \psi) - (\nabla \psi^*) \psi]$$

- determining the differential cross section amounts to solving the Schrödinger equation in order to find  $f(\theta)$

# Scattering experiments

- Most important tool for obtaining detailed information on the structure of matter and the interaction between particles! (*neutron and X-ray scattering, atoms scattering from crystal surfaces, scattering of elementary particles*)
- Beam of incident particles hits a composite target and the distribution of scattered particles over the different directions is measured
- Scattering can be **ELASTIC** (energy is conserved), or **INELASTIC** (energy disappears !)
- **Strategy:** on physical grounds determine the functional form of the potential between the interacting particles. Introduce some parameters which can be obtained by comparing theoretical predictions to experimental data.
- **Inverse scattering problem:** obtain potential from data, does not lead to unique solutions !

## 2.1 Quantum scattering : Introduction I

- Scattering process described by the Schrödinger equation **PDE**

$$\left[ \frac{-\hbar^2}{2m} \nabla^2 + V(r) \right] \Phi(\vec{r}) = E \Phi(\vec{r}) \quad [m: \text{reduced mass}] ,$$

- Solutions can be written as a partial-wave expansion

$$\Phi(\vec{r}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=+l} A_{lm} \frac{u_l(r)}{r} Y_l^m(\theta, \phi) \quad [\Phi(\vec{r}) : \text{dimensionless}]$$

- $u_l(r)$  solution of the **radial Schrödinger equation** **ODE**

$$\frac{\hbar^2}{2m} \frac{d^2}{dr^2} u_l(r) = \left[ V(r) + \frac{\hbar^2 l(l+1)}{2mr^2} - E \right] u_l(r)$$

Remark that  $u_l(r)$  depends on the energy  $E$  ( $u_l(r, E)$ )!

## 2.1 Quantum scattering: Introduction II

- Determine  $r_{max}$  such that  $V(r > r_{max}) \approx 0$ .
- For  $r > r_{max}$ :  $u_l(r > r_{max}) \equiv kr \times f_l(kr = z)$   
with  $f_l(z)$  solution of

$$\left[ \frac{d^2}{dz^2} + \frac{2}{z} \frac{d}{dz} + \left( 1 - \frac{l(l+1)}{z^2} \right) \right] f_l(z) = 0$$

$f_l(z)$  spherical Bessel functions ( $f_l(z) = j_l(z), n_l(z)$ )

$$\lim_{z \gg} z j_l(z) \sim \sin \left( z - \frac{l\pi}{2} \right) \quad \lim_{z \gg} z n_l(z) \sim -\cos \left( z - \frac{l\pi}{2} \right)$$

- From the above one can conclude:

$$u_l(r > r_{max}) \sim \sin \left( kr - \frac{l\pi}{2} + \delta_l \right) \quad \left( E = \frac{\hbar^2 k^2}{2m} \right)$$

- Phase Shift  $\delta_l(E)$ : ENCODES EFFECT OF  $V(r)$

## 2.1 Quantum scattering: Introduction III

### DIFFERENTIAL CROSS SECTION

$$d\Omega \equiv \sin \theta d\theta d\phi$$

$$\frac{d\sigma}{d\Omega} = \frac{1}{k^2} \left| \sum_{l=0}^{l=+\infty} (2l+1) e^{i\delta_l} \sin \delta_l P_l(\cos \theta) \right|^2$$

### TOTAL CROSS SECTION

$$\sigma_{tot} = \int_0^\pi \sin \theta d\theta \int_0^{2\pi} d\phi \frac{d\sigma}{d\Omega} = \frac{4\pi}{k^2} \sum_{l=0}^{l=+\infty} (2l+1) \sin^2 \delta_l$$

- *In practice: Sum over  $l$  can be cut at some  $l_{max}$  (indeed finite amounts of angular momentum are transferred to the target upon interacting with the beam)*
- *Once the phase shifts are known the cross sections can be computed !!*

## 2.1 Quantum scattering: Introduction IV

### HOW TO DETERMINE THE PHASE SHIFTS $\delta_I$ ?

- Solve the radial Schrödinger equation (numerically !) from  $r = 0$  outwards with boundary condition  $u_I(r = 0) = 0$
- Determine  $u_I(r)$  for two values  $r_1$  and  $r_2$  with  $r_1, r_2 > r_{max}$
- Determine phase shift from

$$\tan \delta_I = \frac{K j_I(kr_1) - j_I(kr_2)}{K n_I(kr_1) - n_I(kr_2)} \text{ with, } K = \frac{r_1 u_I(r_2)}{r_2 u_I(r_1)}$$

with  $j_I$  and  $n_I$  the so-called regular and irregular spherical Bessel function !

### This requires

- Numerical routine to solve the radial Schrödinger equation!
- Numerical routines to determine the spherical Bessel functions  $j_I$  and  $n_I$ !

## 2.2 A program for calculating cross sections (p19)

- Particular problem: H scattering off (much heavier) Kr atoms
- Atoms are considered as “structureless” (atoms as point particles)
- Comparison with experimental data (computed cross sections are of great help in interpreting the data!)

2.2.1 Numerov's method for the radial Schrödinger equation

2.2.2 The spherical Bessel function

2.2.3 Putting the pieces together- results

## 2.2.1 Solution of the radial Schrödinger equation

The radial Schrödinger equation adopts the form

$$\frac{\hbar^2}{2m} \frac{d^2}{dr^2} u_l(r) = F(l, r, E) u_l(r) ,$$

with,

$$F(l, r, E) \equiv V(r) + \frac{\hbar^2 l(l+1)}{2mr^2} - E .$$

- **Units** : choose units such that  $\hbar^2/(2m)$  assumes a reasonable value (not extremely large, not extremely small).
- Differential equation can for example be solved with **Runge Kutta method** (*integration step can be varied*), or with **Numerov method** (*simplicity of regular mesh with good efficiency*).

**Solutions  $x(t)$  of the equation**

$$\ddot{x}(t) = f(t) x(t)$$

are given by (order  $O(h^6)$ )

$$x(t) = \frac{w(t)}{\left[1 - \frac{h^2}{12} f(t)\right]}$$

with,

$$w(h) + w(-h) - 2w(0) = h^2 f(0) x(0) + \mathcal{O}(h^6)$$

- Second order integration scheme for  $w(t)$ : use two values of the solution to predict the next one!  $w(-h), w(0) \Rightarrow w(h)$
- **ROUTINE : NUMEROV** (Repeated use results in  $\mathcal{O}(h^4)$  accuracy)

# Proof of Numerov's algorithm (I)

- For an arbitrary function  $w(t)$  one can write:

$$w(h) + w(-h) = 2w(0) + h^2 \ddot{w}(0) + 2 \frac{h^4}{4!} \dddot{w}(0) + \mathcal{O}(h^6) \quad (NUM1)$$

- Special situation:

$$\begin{cases} w(t) & \equiv x(t) \left[ 1 - \frac{h^2}{12} f(t) \right] \\ \ddot{x}(t) & = f(t)x(t) \end{cases}$$

- Since

$$\dddot{x}(t) = \frac{d^2}{dt^2} \ddot{x}(t) = \frac{d^2}{dt^2} x(t)f(t)$$

One has:

$$\ddot{w}(t) = \ddot{x}(t) - \frac{h^2}{12} \dddot{x}(t) \quad (NUM2)$$

## Proof of Numerov's algorithm (II)



$$h^4 \ddot{w}(t) = h^4 \ddot{x}(t) + \mathcal{O}(h^6) \quad (NUM3)$$

- Insert (NUM2) and (NUM3) in (NUM1) to obtain that

$$w(h) + w(-h) - 2w(0) = h^2 f(0) x(0) + \mathcal{O}(h^6)$$

## Numerov's method applied to radial Schrödinger Eq.

$$w_I(r + h) = 2w_I(r) - w_I(r - h) + h^2 F(I, r, E) u_I(r)$$

with,

$$\frac{\hbar^2}{2m} \equiv 1 \quad F(I, r, E) = V(r) + \frac{I(I+1)}{r^2} - k^2 (= E)$$
$$u_I(r) = \frac{w_I(r)}{1 - \frac{\hbar^2}{12} F(I, r, E)}$$

- The starting value at  $r = 0$  for a “regular” potential  $\left(\lim_{r \rightarrow 0} F(I, r, E) = \frac{I(I+1)}{r^2}\right)$ :  $u_I(r = 0) = w_I(r = 0) = 0$
- The starting value at  $r = h$  for a “regular” potential :  $u_I(r = h) = h^{I+1}$
- For a Lennard-Jones potential (which diverges strongly near the origin): different boundary conditions must be used!

# FORTRAN implementation of Numerov algorithm (I)

```
SUBROUTINE Numerov (Delta, StartI, EndI, MaxSol, FArr,  
 . . . Sing, PhiStart, PhiNext, Solution)  
C Integrates the Schrodinger equation. The initial values  
C of the solution are PhiStart and PhiNext. The integration  
C step is Delta, and the integration steps run from StartI to EndI.  
C StartI may be larger than EndI : integration is performed backward.  
C The output values is the solution, stored in the array "Solution".  
C Sing determines whether the potential contains a singularity at  
C r=0.  
C If there is a singularity at r=0, the value of the Numerov  
C function w at r=0 is taken equal to PhiStart, and not  
C equal to PhiStart/(1-h^2 FArr/12).  
  
C This array is declared with linear size MaxSol.  
C Delta is the integration step.  
C The equation solved is  
C Psi''(R_I) = FArr(I) Psi(R_I)  
C FArr must therefore be filled with the appropriate values before  
C calling the present routine. In the case of the radial  
C Schrodinger equation, FArr would contain the values  
C FArr(I) = 2*(V(R)-E)+L*(L+1)/R**2 for R=R_I.  
IMPLICIT NONE  
INTEGER I, StartI, EndI, MaxSol, IStep  
DOUBLE PRECISION Phi, PhiStart, PhiNext, W, WNext, WPrev,  
 . . . Delta, Fac, FArr(MaxSol), DeltaSq, Solution(MaxSol)  
LOGICAL Sing
```

# FORTRAN implementation of Numerov algorithm (II)

```
IF (Delta.LT.0) THEN
    IStep = -1
ELSE
    IStep = 1
ENDIF
DeltaSq = Delta*Delta
Fac = DeltaSq/12.D0

IF (Sing) THEN
    WPrev = PhiStart
ELSE
    WPrev = (1-Fac*FArr(StartI))*PhiStart
    Solution(StartI) = PhiStart
ENDIF

Phi = PhiNext
Solution(StartI+IStep) = PhiNext
W = (1-Fac*FArr(StartI+IStep))*Phi

DO I = StartI+IStep, EndI-IStep, IStep
    WNext = W*2.D0 - WPrev + DeltaSq*Phi*FArr(I)
    WPrev = W
    W = WNext
    Phi = W/(1-Fac*FArr(I+IStep))
    Solution(I+IStep) = Phi
ENDDO
END
```

# PYTHON implementation of Numerov algorithm (I)

```
def Numerov( Delta , Startl , Endl , FArr , Singular , Phistart , Phinext):  
# Integrates the Schrodinger equation. The initial values  
# of the Solution are Phistart and Phinext respectively.  
# The integration step is Delta  
# The integration steps run from Startl to Endl. Startl may be larger  
# than Endl; in that case, integration is performed backward.  
# The output values is the Solution, stored in the array "Solution".  
# Sing determines whether the potential contains a singularity at  
# r=0.  
# If there is a singularity at r=0, the value of the Numerov  
# function w at r=0 is taken equal to Phistart, and not  
# equal to Phistart/(1-h^2 FArr/12).  
#  
# This array is declared with linear size MaxSol.  
# Delta is the integration step.  
# The equation solved is  
#  $\Psi''(R_I) = FArr(I) \Psi(R_I)$   
# FArr must therefore be filled with the appropriate values before  
# calling the present routine. In the case of the radial  
# Schrodinger equation, FArr would contain the values  
#  $FArr(I) = 2*(V(R)-E)+L*(L+1)/R^2$  for  $R=R_I$ .
```

# PYTHON implementation of Numerov algorithm (II)

```
Solution = [None]*MaxSol
if Delta > 0:
    IStep = 1
else:
    IStep = -1
Deltasq = Delta*Delta
Fac = Deltasq/12.
if( Singular):
    Wprev = Phistart
else:
    Wprev=(1-Fac*FArr[ StartI+IStep ])*Phistart
    Solution[ StartI ] = Phistart
Phi = Phinext
Solution[ StartI+IStep]= Phinext
W = ( 1 - Fac * FArr[ StartI+IStep ] )*Phi
for i in range( StartI+IStep , EndI-IStep+1, IStep ):
    Wnext = W*2. - Wprev + Deltasq*Phi+FArr[ i ]
    Wprev = W
    W      = Wnext
    Phi    = W(1-Fac*FArr[ i+IStep ])
    Solution[ i+IStep ] = Phi
return Solution
```

# How to test programs ??

Essential to build a program step by step and to check every routine extensively !

- comparison with analytical solutions!
- comparison with tabulated values

For example: testing numerical routines for special functions:

- 1 *"Handbook of Mathematical Functions"*, Milton Abramowitz and Irene A. Stegun (Dover Publications, 1970).

This book is available on line

[http://en.wikipedia.org/wiki/Abramowitz\\_and\\_Stegun](http://en.wikipedia.org/wiki/Abramowitz_and_Stegun)

- 2 NIST digital library of Mathematical Physics: <http://dlmf.nist.gov>

- look at trends! For example, as the energy  $E$  increases in the energy continuum of a potential, the  $u_l(r)$  should look more and more like a plane wave!

# Test of Numerov method with 3-dimensional isotropic HO (Check 1 on page 20)

- Choose units such that  $\frac{\hbar^2}{2m} = 1$  and choose  $\frac{k}{2} = 1$ .
- The energy eigenvalues

$$E = \hbar\omega \left( n + \frac{3}{2} \right) = 2 \left( n + \frac{3}{2} \right) \quad (n = 0, 1, 2, 3, \dots),$$

- Solutions of the radial Schrödinger equation

$$\frac{u_{El}(r)}{r} = N e^{-\alpha^2 r^2/2} r^I {}_1F_1 \left( \frac{I-n}{2}, I + \frac{3}{2}, \alpha^2 r^2 \right),$$

- The solution for  $(n = 0, I = 0)$  reads  $u(r) = A r e^{-r^2/2}$

- 

$$F(I, r, E) = r^2 + \frac{I(I+1)}{r^2} - E$$

## 2.2.2 Spherical Bessel Function (A2, p558-559) I

### ITERATIVE PROCEDURES FOR SPECIAL FUNCTIONS

- Upward recursion formula: ( $s_l(x) \equiv j_l(x)$  or,  $n_l(x)$ )

$$s_{l+1}(x) = \frac{2l+1}{x} s_l(x) - s_{l-1}(x)$$

- The  $j_l(x)$  are regular for  $x = 0$

$$j_0(x) = \frac{\sin x}{x} \text{ and, } j_1(x) = \frac{\sin x}{x^2} - \frac{\cos x}{x}$$

- The  $n_l(x)$  are irregular for  $x = 0$  (singular)

$$n_0(x) = \frac{-\cos x}{x} \text{ and, } n_1(x) = \frac{-\cos x}{x^2} - \frac{\sin x}{x}$$

- Knowledge of  $j_0(x)$  and  $j_1(x)$  allows to determine  $j_2$  and so on.

ROUTINE: **SPHBESJ** AND **SPHBESN**

## 2.2.2 Spherical Bessel Function (A2, p558-559) II

- For some  $(l, x)$  combinations: upward recursion suffers from “subtractive cancellation” (difference between two large numbers is prone to round-off errors)
- Three-point recursion formula: two independent solutions  $(j_l, n_l)$

$$j_l^{\text{computed}}(x) = j_l^{\text{real}}(x) + \epsilon n_l^{\text{real}}(x)$$

$\epsilon$  is due to the computer's lack of precision!

- One of these two solutions may grow strongly with  $l$
- If solution we are after damps out and the other one grows with  $l$ : even for small values of  $\epsilon$  large errors will sneak in the solution (due to finite precision of computer!)
- Types of error: Blunders or bad theory, random errors (rare), approximation errors (simplification of the mathematics), round-off errors (finite number of digits to store numbers)

## 2.2.2 Spherical Bessel Function (A2, p558-559) III

- Problems can be avoided by performing the recursion downward

$$s_{l-1}(x) = \frac{2l+1}{x} s_l(x) - s_{l+1}(x)$$

$$s_{l_{top}+1}(x), s_{l_{top}}(x) \rightarrow s_{l_{top}-1}(x)$$

- Start with arbitrary values:  $s_{l_{top}+1}(x) = 0, s_{l_{top}}(x) = \delta$  ( $\delta$  small)
- Normalisation: continue iteration down to  $l = 0, l = 1$

$$j_l(x) = j_l^{\text{computed}}(x) \times \frac{j_0(x)}{j_0^{\text{computed}}(x)}$$

- Problem: how large should  $l_{top}$  be? (Often: asymptotic expressions from functions)

# Python Routine to determine $j_l(r)$

```
import math as m
def SphBesJ(L, X):
    # Returns the spherical bessel function  $j_l(x)$  as a function of  $l$  and  $x$ 
    # Upward recursion is used; therefore reliable for a restricted number of
    #  $l$ -values
    if L==0:
        return m.sin(X)/X
    elif L==1:
        return m.sin(X)/X-X-m.cos(X)/X
    else:
        HelpSin = m.sin(X)
        JLMin1 = HelpSin/X
        JL = JLMin1/X-m.cos(X)/X
        for HelpL in range(2, L+1):
            JLMin2 = JLMin1
            JLMin1 = JL
            JL = (2*HelpL-1)/X*JLMin1 - JLMin2
        return JL
```

# Python routine to determine $n_l(r)$

```
import math as m
def SphBesN( L, X):
#
# Returns the spherical bessel function n_l(x) as a function of l and x
# Upward recursion is used.
if L==0:
    return -m.cos(X)/X
elif L==1:
    return -m.cos(X)/X-X-m.sin(X)/X
else:
    HelpCos = m.cos(X)
    NLMin1 = -HelpCos/X
    NL = NLMin1/X-m.sin(X)/X
    for HelpL in range(2, L+1):
        NLMin2 = NLMin1
        NLMin1 = NL
        NL = (2*HelpL-1)/X*NLMIn1 - NLMin2
    return NL
```

## 2.2.3 An example: Hydrogen-Krypton scattering (I)

- Molecular H-Kr interactions: Lennard-Jones potential

$$V_{LJ}(r) = \epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - 2 \left( \frac{\sigma}{r} \right)^6 \right]$$

with  $\epsilon = 5.9 \text{ meV}$ ,  $\sigma = 3.57 \times 10^{-10} \text{ m}$  two parameters.

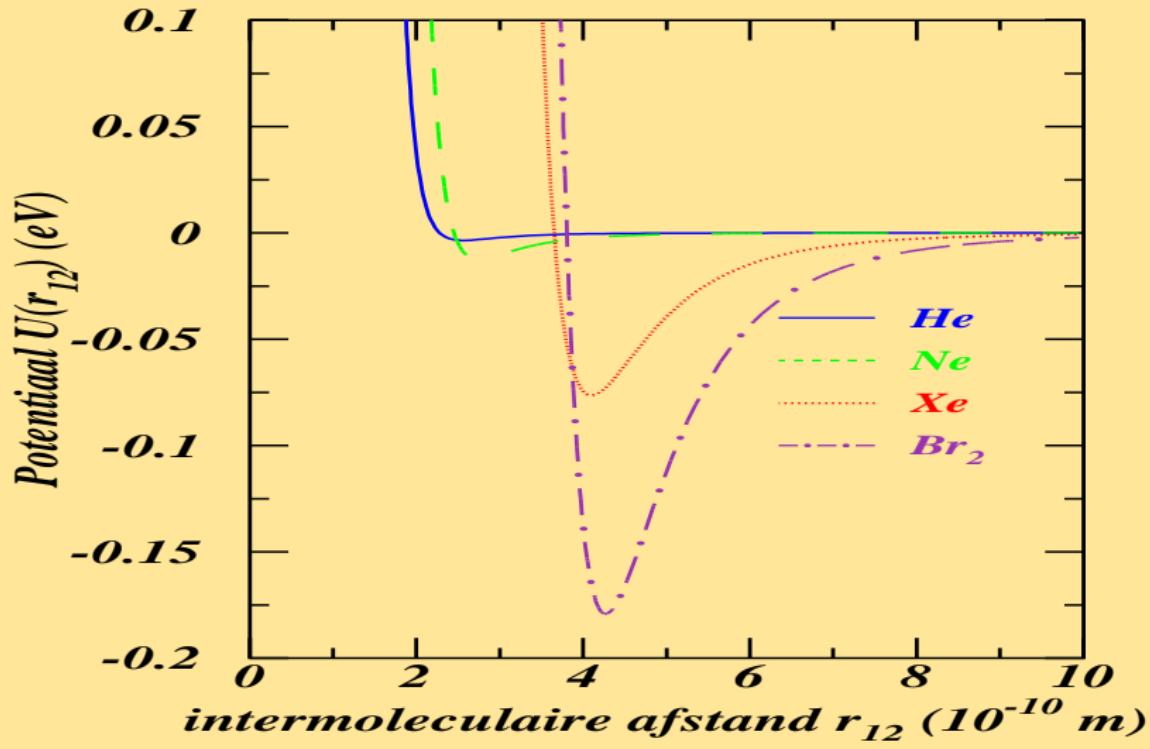
- At sufficiently small values of  $r$  ( $r < r_{min}$ )

$$\frac{d^2}{dr^2} u_I(r) = \frac{2m\epsilon\sigma^{12}}{\hbar^2} \frac{u_I(r)}{r^{12}}$$

$$\epsilon \left( \frac{\sigma}{r} \right)^{12} \gg \left[ 2\epsilon \left( \frac{\sigma}{r} \right)^6, E, \frac{\hbar^2 I(I+1)}{2mr^2} \right]$$

SOLUTION:  $u_I(r) = \exp - \left( \sqrt{\frac{\epsilon\alpha}{25}} \frac{1}{r^5} \right)$  with,  $\alpha = \frac{2m\sigma^{12}}{\hbar^2}$

## Intermoleculaire Lennard-Jones Potentiaal



## An example : Hydrogen-Krypton scattering (II)

- Unit of distance :  $\sigma$  (a few  $10^{-10}$  m)
- Unit of energy : meV
- In these units :

$$\frac{2m}{\hbar^2} \approx 6.12(\text{meV})^{-1} \cdot \sigma^{-2}$$

- Effective potential ( $V_{\text{eff}}(I, r) = F(I, r, E = 0)$ )

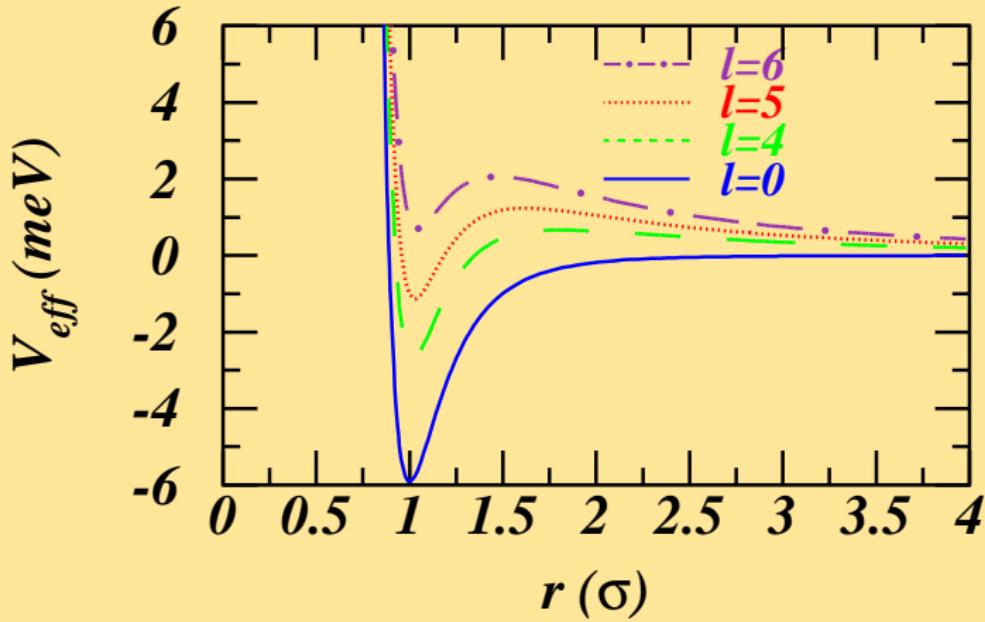
$$V_{\text{eff}}(I, r) = V_{\text{LJ}}(r) + \frac{\hbar^2 I(I+1)}{2mr^2}$$

- In terms of  $V_{\text{eff}}(r)$  the radial Schrödinger equation can be rewritten as :

$$-\frac{\hbar^2}{2m} \frac{d^2}{dr^2} u_I(r) + V_{\text{eff}}(I, r) u_I(r) = E u_I(r) .$$

This equation looks like an eigenvalue equation !

# Effective Lennard-Jones Potentiaal



For higher  $l$ : quasi-bound states are possible !

Program to compute total  $H - Kr$  cross sections

- Routine `InputParams` initializes the calculation
- Routine `CalcScatter` computes the total cross sections

## The routine `InputParams`

- Input: the range of energies for which the cross sections are computed (variables `MaxEner`, `MinEner`, `EnerNum`)
- Input:  $l_{max}$  in the partial-wave expansion (variable `LMax`)
- Input: the ranges ( $r_{min}$  and  $r_{max}$ ) and the step-width ( $h$ ) for the solution of the radial Schrödinger equation (variables `Start`, `MaxDist`, `HStep`)

## The routine CalcScatter

- perform a loop over the different energies  $E$
- for each energy perform a loop over the partial waves  $(0 \leq l \leq l_{max})$
- for each partial wave  $l$  and energy  $E$  determine the phase shift  $\delta_l$  (by making use of the routine CalcDelta)
- Cross section at a fixed value of  $E$  can be obtained by summing over the different partial waves  $l$

# Routine CalcDelta

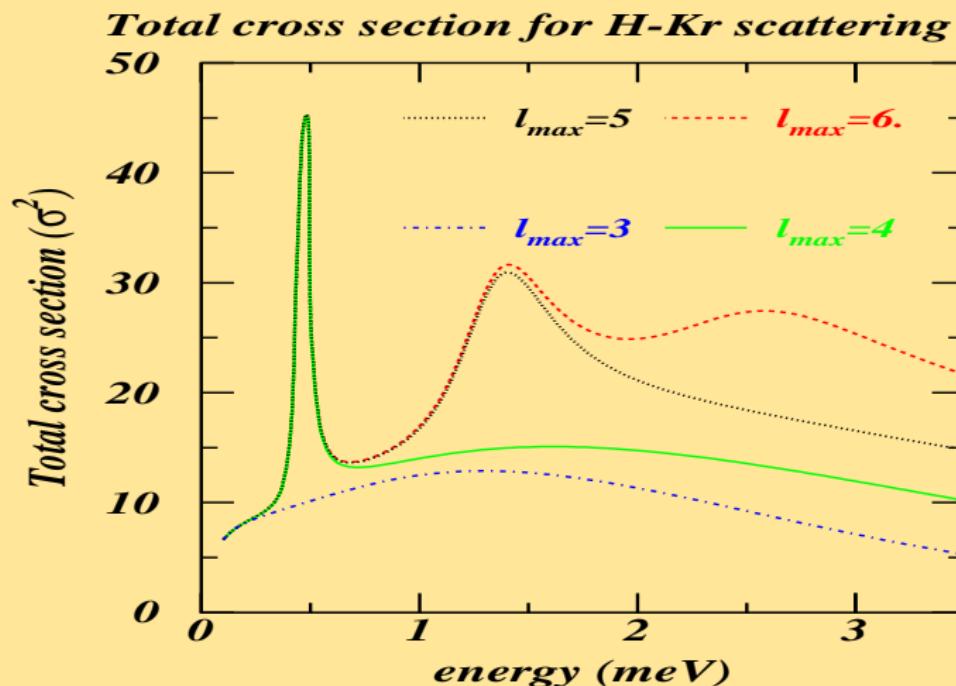
The routine `CalcDelta` determines the phase shifts!

## The routine `CalcDelta`

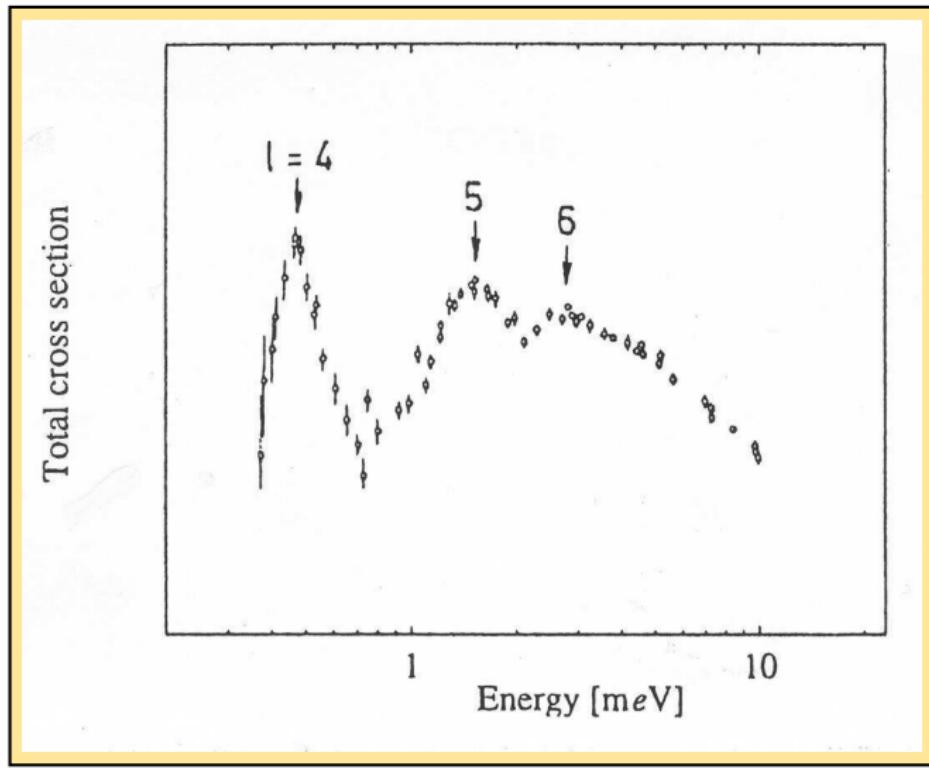
- Call to the routine `FindGQuotient`
  - solve the radial Schrödinger equation for a given partial wave  $l$  and energy  $E$  (by making use of the routines `FillArr` and `Numerov`)
  - determine  $u_l(r, E)$  for  $r_1, r_2 > r_{max}$  (variables `PhiEnd1` and `PhiEnd2`)
  - variable `GQuotient` =  $K$
- Call to the routine `PhaseShift`: determines the phaseshift  $\delta_l$  from the knowledge of  $K$

$$K = \frac{r_1 u_l(r_2, E)}{r_2 u_l(r_1, E)}$$

# Computed total H-Kr cross sections



# Experimental results for H-Kr scattering



J.P. Toennies *et al.*, J. Chem. Phys. **71** (1979) 614.

## 2.3 Calculation of scattering cross sections (I)

- Ansatz for sufficiently large value of  $r$  ( $r \geq r_{max}$ )

$$\text{ANSATZ 1: } \Phi(\vec{r}) = e^{i\vec{k} \cdot \vec{r}} + f(\theta) \frac{e^{ikr}}{r} = \Phi_i + \Phi_f$$

$f(\theta)$  : scattering amplitude (*not dependent on  $\phi$  due to spherical symmetry*)

- Differential cross section is given by:

$$\frac{d\sigma}{d\Omega} = \frac{\text{Scattered energy / solid angle}}{\text{incident energy}} = \frac{r^2 |\Phi_f|^2}{|\Phi_i|^2} = |f(\theta)|^2 .$$

- For  $r \geq r_{max}$  one can perform partial-wave expansion leaving out all  $m \neq 0$  contributions (would introduce  $\phi$  dependence):

$$\text{ANSATZ 2: } \Phi(\vec{r}) = \sum_{l=0}^{l=+\infty} A_l \frac{u_l(r)}{r} P_l(\cos \theta)$$

## 2.3 Calculation of scattering cross sections (II)

Two expressions for the asymptotic wave function  $\Phi(\vec{r})$  are now available.

- Introduce the expansion of a plane wave ( $\theta$  is the angle between  $\vec{k}$  and  $\vec{r}$ ):

$$e^{i\vec{k}\cdot\vec{r}} = \sum_{l=0}^{l=\infty} i^l (2l+1) j_l(kr) P_l(\cos \theta) .$$

- Introduce an expansion for the scattering amplitude:

$$f(\theta) = \sum_{l=0}^{l=+\infty} f_l P_l(\cos \theta) .$$

- Asymptotic expression for a spherical Bessel function:

$$j_l(kr) \approx \frac{\sin(kr - \frac{l\pi}{2})}{kr} \approx \frac{e^{i(kr - \frac{l\pi}{2})} - e^{-i(kr - \frac{l\pi}{2})}}{2ikr} \quad (r \geq r_{max})$$

## 2.3 Calculation of scattering cross sections (III)

- Asymptotic expression for the radial wave functions ( $r \geq r_{max}$ ):

$$u_l(r) \approx \sin \left( kr - \frac{l\pi}{2} + \delta_l \right).$$

- The two asymptotic forms for  $\Phi(\vec{r})$  contain incoming and outgoing spherical waves. For each partial wave  $l$  the prefactors of the outgoing and incoming spherical waves should be equal:

$$A_l = (2l+1)i^l e^{i\delta_l} \quad \text{en,} \quad f_l = \frac{2l+1}{k} e^{i\delta_l} \sin \delta_l.$$

- Accordingly:

$$\frac{d\sigma}{d\Omega} = |f(\theta)|^2 = \left| \sum_{l=0}^{l=+\infty} f_l P_l(\cos \theta) \right|^2$$

# Chapter 3 : The variational method for the Schrödinger equation

- 3.1 Variational calculus
- 3.2 Examples of variational calculations
  - 3.2.1 The infinitely deep potential well
  - 3.2.2 Variational calculation for the hydrogen atom
- 3.3 Solution of the generalised eigenvalue equation
- 3.4 Perturbation theory and the variational method
- Example: The finite potential well (Exercise 3.4 on page 43)

### 3.1 Variational calculus (I)

- Analytical solutions to the Schrödinger equation: restricted number of cases.
- Numerical solutions to the Schrödinger equation through integration: huge number of grid points! (Section A.7.2)
- Numerical solutions to the Schrödinger equation with the **VARIATIONAL METHOD**: often very efficient!

Define the **ENERGY FUNCTIONAL**

$$E[\phi] = \frac{\int dX \phi^\dagger \mathcal{H} \phi}{\int dX \phi^\dagger \phi} = \frac{\langle \phi | \mathcal{H} | \phi \rangle}{\langle \phi | \phi \rangle} = \frac{P}{Q}$$

with e.g.,  $dX \equiv (d\vec{r}_1, d\vec{r}_2, \dots, d\vec{r}_n, d\vec{p}_1, d\vec{p}_2, \dots, d\vec{p}_n, d\vec{\sigma}_1, d\vec{\sigma}_2, \dots, d\vec{\sigma}_n)$

$\mathcal{H}\phi = E\phi$  equivalent with,  $\delta E[\phi] = 0$

### 3.1 Variational calculus (II)

- **STATIONARY STATES  $\phi$ :** for  $\phi \rightarrow \phi + \delta\phi$  the corresponding change in  $E[\phi]$  vanishes to first order!

$$\delta E[\phi] = 0 = E[\phi + \delta\phi] - E[\phi] = \frac{\langle \phi + \delta\phi | \mathcal{H} | \phi + \delta\phi \rangle}{\langle \phi + \delta\phi | \phi + \delta\phi \rangle} - \frac{P}{Q}$$

- Retain only terms linear in  $\delta\phi$  to obtain that:

$$\delta E[\phi] = 0 = \frac{\langle \delta\phi | \mathcal{H} - \frac{P}{Q} | \phi \rangle + \langle \phi | \mathcal{H} - \frac{P}{Q} | \delta\phi \rangle}{Q + \langle \phi | \delta\phi \rangle + \langle \delta\phi | \phi \rangle}$$

- As this should be valid for an arbitrary small change in  $\phi$

$$\mathcal{H}\phi = \frac{P}{Q}\phi = E\phi \quad (+\text{Hermitian conjugate})$$

- **VARIATIONAL CALCULUS:** stationary states of  $E[\phi]$  are found within a subspace of the full Hilbert space!

### 3.1 Variational calculus (III)

**Linear variational calculus** within subspace of the Hilbert space

$$|\phi\rangle = \sum_{p=1}^{p=N} C_p |\chi_p\rangle$$

with  $\chi_p$ , set of well-chosen, continuous, analytic, (orthonormal) basis vectors! The stationary states and eigenvalues follow from

$$\sum_{q=1}^{q=N} (H_{pq} - E\delta_{pq}) C_q = 0, \text{ for, } p = 1, 2, \dots, N$$

$$H_{pq} \equiv \langle \chi_p | \mathcal{H} | \chi_q \rangle$$

**Eigenvalue equation  $\mathbf{H} \mathbf{C} = E \mathbf{C}$**  for a finite, orthonormal basis:

solves the Schrödinger equation !

Non-linear variational calculus is far more complex!!

### 3.1 Variational calculus (IV)

- **Generalised eigenvalue equation  $\mathbf{H} \mathbf{C} = E \mathbf{S} \mathbf{C}$**  with  $S_{pq} = \langle \chi_p | \chi_q \rangle$   
*Computer time needed for matrix diagonalisation scales with the third power of the dimension ( $\mathcal{O}(N^3)$  process)*
- Standard discretisation methods (like finite difference) are  $\mathcal{O}(N)$
- Set of basis functions  $\chi_p$  must be chosen carefully so as to approximate the full solution with  $N$  as small as possible!
- The lowest energy eigenvalue is always greater than or equal to the “real” ground-state energy!
- An excellent set of FORTRAN (77 +90), routines for matrix manipulations: LAPACK (*Linear Algebra PACKAGE*) from <http://www.netlib.org/>.
- Routines can be linked with C and C++ programs
- PYTHON: scipy's `scipy.linalg.eigh` for generalized eigenvalue problems for symmetric (hermitian) positive definite matrices
- A generalised eigenvalue equation can be transformed into an ordinary eigenvalue equation

### 3.3 Solution of the generalised eigenvalue problem

$$\mathbf{H} \mathbf{C} = E \mathbf{S} \mathbf{C}$$

- 1 Step 1: find matrix  $\mathbf{V}$  which brings matrix  $\mathbf{S}$  to unit form

$$\mathbf{V}^\dagger \mathbf{S} \mathbf{V} = \mathbf{I}$$

- 2 Step 2: rewrite the generalised eigenvalue problem as

$$\mathbf{V}^\dagger \mathbf{H} \mathbf{V} \mathbf{V}^{-1} \mathbf{C} = E \mathbf{V}^\dagger \mathbf{S} \mathbf{V} \mathbf{V}^{-1} \mathbf{C}$$

- 3 Step 3: define two matrices

$$\begin{aligned}\mathbf{V}^\dagger \mathbf{H} \mathbf{V} &= \mathbf{H}' \\ \mathbf{V}^{-1} \mathbf{C} &= \mathbf{C}'\end{aligned}$$

- 4 Step 4: with these definitions one has  $\mathbf{H}' \mathbf{C}' = E \mathbf{C}'$

- 5 Step 5: Solve for  $E$  and  $\mathbf{C}'$  and determine eigenvectors of the original problem as  $\mathbf{C} = \mathbf{V} \mathbf{C}'$

### 3.2.1 The infinitely deep potential well (I)

$$V(x) = \begin{cases} +\infty & \text{for } |x| > 1 \\ = 0 & \text{for } |x| \leq 1 \end{cases}$$

- Units:  $\frac{\hbar^2}{2m} = 1$
- Basis functions ( $\psi_n(x = \pm 1) = 0$ )

$$\psi_n(x) = x^n(1-x)(1+x), \quad n = 0, 1, 2, \dots$$

- Overlap matrix ( $n + m$  even)  $\left[ S_{nm} = 0 \text{ for } (n + m) \text{ odd} \right]$

$$S_{nm} = \langle \psi_n | \psi_m \rangle = \frac{2}{n+m+5} - \frac{4}{n+m+3} + \frac{2}{n+m+1}.$$

- Hamiltonian matrix  $\left[ H_{nm} = 0 \text{ for } (n + m) \text{ odd} \right]$

$$H_{nm} = \int_{-1}^{+1} dx \psi_n(x) \left( -\frac{d^2}{dx^2} \right) \psi_m(x)$$

### 3.2.1 The infinitely deep potential well (II)

#### EXACT SOLUTIONS:

- Standing wave conditions  $n\frac{\lambda}{2} = L$  (here,  $L = 2$ )
- Eigenfunctions are standing waves

$$\sin\left(\frac{n\pi x}{L}\right) \implies \sin\left(\frac{n\pi x}{2}\right) \quad n \text{ even and positive}$$
$$\cos\left(\frac{n\pi x}{L}\right) \implies \cos\left(\frac{n\pi x}{2}\right) \quad n \text{ odd and positive}$$

- Corresponding eigenvalues

$$E_n = \frac{\hbar^2 k^2}{2m} \implies E_n = \frac{n^2 \pi^2}{4} \quad (\text{for } \frac{\hbar^2}{2m} \equiv 1)$$

### 3.2.1 The infinitely deep potential well (III)

Energy eigenvalues from a variational calculation in a basis of dimension  $N$

36

*The variational method for the Schrödinger equation*

Table 3.1: Energy levels of the infinitely deep potential well. The first four columns show the variational energy levels for various numbers of basis states  $N$ . The last column shows the exact values. The exact levels are approached from above as in figure 3.1.

$N = 5$	$N = 8$	$N = 12$	$N = 16$	Exact
2.4674	2.4674	2.4674	2.4674	2.4674
9.8754	9.8696	9.8696	9.8696	9.8696
22.2934	22.2074	22.2066	22.2066	22.2066
50.1246	39.4892	39.4784	39.4784	39.4784
87.7392	63.6045	61.6862	61.6850	61.6850

Computed eigenvalues in a limited Hilbert space are always larger than the exact ones! (*thermodynamic principle: physical eigenstates @  $T = 0$  are those with the lowest energy!*)

# Program DEEPWELL (I)

Program to compute energy eigenvalues and eigenfunctions in an infinitely deep potential well

- Input: the dimension of the basis (variable  $N$ )
- Routine `CalcSpectrum` computes energy eigenvalues and eigenfunctions

## The routine `CalcSpectrum`

- Input: the definition of the basis functions
- Input: the dimension of the basis (variable  $N$ )
- Fill the matrices  $H$  and  $S$  (variables `HMatrix(I,J)`, `SMatrix(I,J)`)
- Call to the routine `GenEig` (solves generalised eigenvalue problem)
- Output: the eigenfunctions (variable `Res(I,J)`) and eigenvalues (variable `Diag(I)`)

# Program DEEPWELL (II)

## The routine Geneig

- Input:  $HMatrix(I,J)$ ,  $SMatrix(I,J)$
- Input: the dimension of the basis (variable  $N$ )
- Call to the LAPACK routine `DSYGV` (solves generalised eigenvalue problem)
- Output: the eigenfunctions (variable  $Res(I,J)$ ) and eigenvalues (variable  $Diag(I)$ )

The name of each LAPACK routine is a coded specification of its function **XXYYZZZ**.

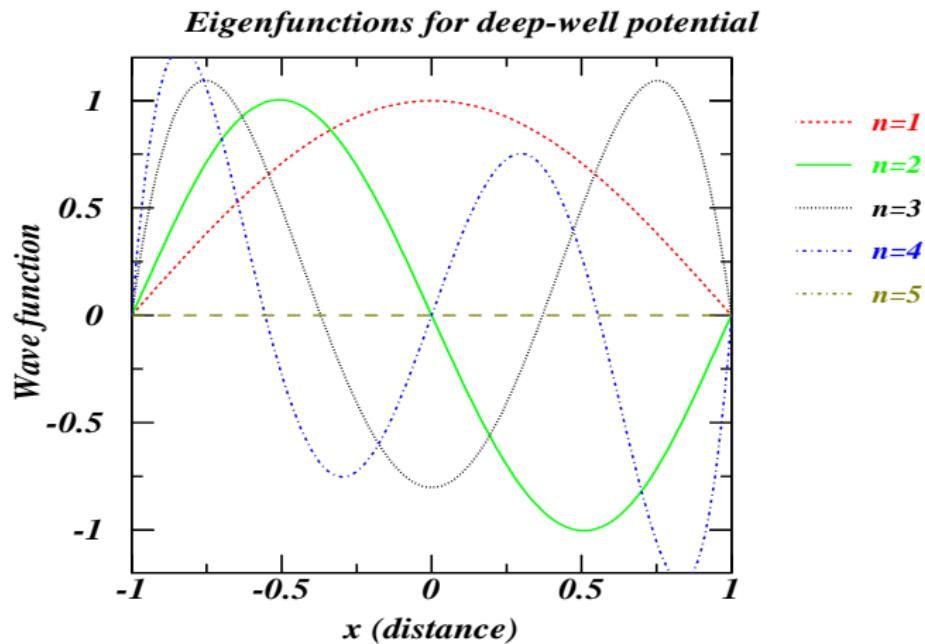
- The first letter **X** indicates the data type:  $X = S$ : REAL ;  $X = D$ : DOUBLE PRECISION ;  $X = C$ : COMPLEX and  $X = Z$ : DOUBLE COMPLEX
- The next two letters **YY** indicate the type of matrix. For example:  $YY = DI$ : diagonal ;  $YY = HE$ : Hermitian ;  $YY = SY$ : symmetric.
- The last three letters **ZZZ** indicate the operation performed.

# Program DEEPWELL: PYTHON IMPLEMENTATION

```
from scipy.linalg import eigh
import numpy as np
#####
# build Hamiltonian matrix H and Overlap matrix S for infinite potential well
# and solve the generalized eigenvalue equation
def CalcSpectrum(nf):
    HMatrix = np.zeros((nf,nf),dtype=np.float64)
    SMatrix = np.zeros((nf,nf),dtype=np.float64)
    for m in np.arange(0,nf,1):
        for n in np.arange(0,nf,1):
            if ( (m+n) % 2 == 0): # check whether (m+n) is even
                HMatrix[m,n] = -8.*(1.-m-n-2.*m*n)/((m+n+3.)*(m+n+1.)*(m+n-1.))
                SMatrix[m,n] = +2./(n+m+5.) - 4./(n+m+3.) + 2./(n+m+1.)
            else:
                HMatrix[m,n]=0.
                SMatrix[m,n]=0.
    #
    # Solve generalized eigenvalue equation: eigh(HMatrix ,SMatrix)
    # calculate the eigenvalues w and eigenvectors v
    # HMatrix , Smatrix have dimensions nf x nf
    # w[i]: dimension nf — v[ ij ][ j]: dimension nf x nf
    # v[i][j]: expansion coefficient i corresponding with eigenvalue j
    w,v = eigh(HMatrix ,SMatrix)
    return w,v
#####
n = int(raw_input("[Input]: How many basis functions for INFINITELY DEEP WELL? "))
# get eigenvalues/vectors for the infinite well with x*(x-1)*(x+1) basisf
Diag,Res = CalcSpectrum(n)
print "\n[Output:] Energy Eigenvalues"
for jj in np.arange(0,n,1):
    print "n: %4d" % int(jj+1), "Exact: %10.4f" % ((jj+1.)*np.pi/2.)**2, "Computed: %10.4f" % Diag
#
```

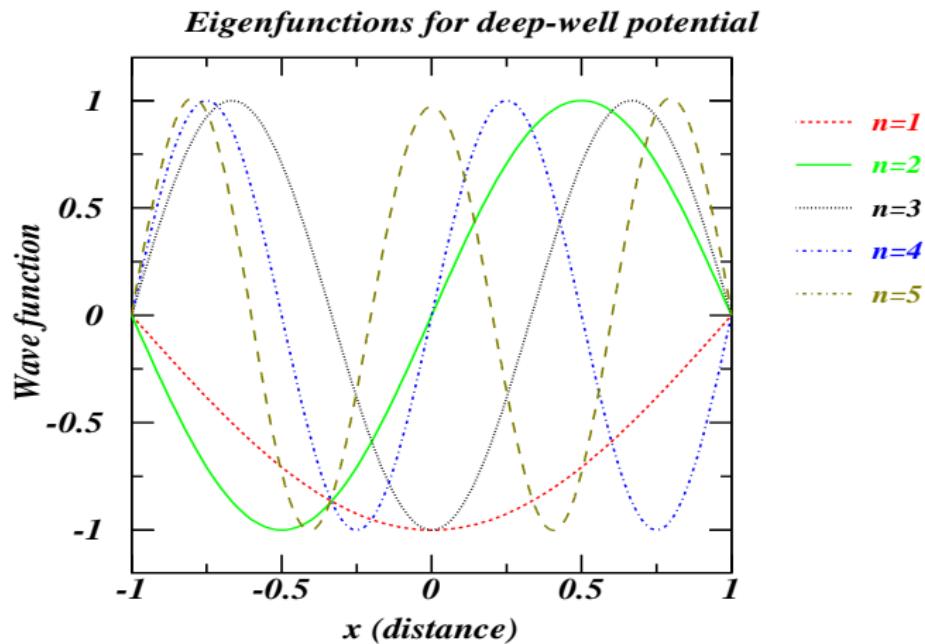
### 3.2.1 The infinitely deep potential well (IV)

Lowest 5 eigenfunctions as computed with  $N=4$



### 3.2.1 The infinitely deep potential well (V)

Lowest 5 eigenfunctions as computed with  $N=10$



### 3.2.2 Variational calculation for hydrogen atom (I)

- Electron-electron interactions (at the origin of the electron-electron correlations) are difficult to treat!
- For hydrogen: direct variational treatment is possible!
- Electronic Schrödinger equation for the hydrogen atom

$$\left[ -\frac{\hbar^2}{2m} \nabla^2 - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r} \right] \Psi(\vec{r}) = E\Psi(\vec{r})$$

- Exact ground-state energy eigenvalue and wave function

$$\Psi(\vec{r}) = \frac{2}{a_0^{3/2}} \exp\left(-\frac{r}{a_0}\right) \text{ and, } E = -13.6058 \text{ eV}$$

$a_0$  : Bohr radius ( $a_0=0.52918 \times 10^{-10} \text{ m}$ )

# Units in electronic structure calculations (ATOMIC UNITS)

## ■ UNIT OF DISTANCE: BOHR RADIUS

$$a_0 = \frac{4\pi\epsilon_0\hbar^2}{m_e e^2} = 0.529189379 \times 10^{-10} \text{ m}.$$

## ■ UNIT OF MASS: ELECTRON MASS

$$m_e = 9.2095 \times 10^{-31} \text{ kg} = 0.51100 \frac{\text{MeV}}{c^2}$$

## ■ UNIT OF CHARGE: ELECTRON CHARGE

$$e = 1.6022 \times 10^{-19} \text{ C}$$

## ■ UNIT OF ENERGY: HARTREE

$$E_H = m_e c^2 \times \alpha^2 = m_e c^2 \times \left( \frac{e^2}{4\pi\epsilon_0\hbar c} \right)^2 = 27.212 \text{ eV}$$

### 3.2.2 Variational calculation for hydrogen atom (II)

- In computer programming, convenient to use units such that equations take on a simple form, not involving extremely large or small constants

$$\left[ -\frac{1}{2} \nabla^2 - \frac{1}{r} \right] \Psi(\vec{r}) = E \Psi(\vec{r}) \quad (\text{with } E \text{ expressed in Hartree units})$$

- Try to approximate the ground-state energy and wave function of the hydrogen atom in a linear variational procedure (only  $l = 0$  wave functions need to be included!)
- Gaussian basis functions are extremely popular in electronic structure calculations (Gaussian integrals!)

$$\chi_p(r) = \exp -\alpha_p r^2$$

- Determine  $C_p$  and  $\alpha_p$ : nonlinear variational problem!

### 3.2.2 Variational calculation for hydrogen atom (III)

[http://en.wikipedia.org/wiki/Gaussian\\_integral](http://en.wikipedia.org/wiki/Gaussian_integral)

- Solving the generalized eigenvalue problem: determine the overlap matrix  $\mathbf{S}$  and the Hamiltonian matrix  $\mathbf{H}$

$$S_{pq} = \int d\vec{r} e^{-\alpha_p r^2} e^{-\alpha_q r^2} = \left( \frac{\pi}{\alpha_p + \alpha_q} \right)^{3/2}$$

$$H_{pq} = T_{pq} \text{ (kinetic energy)} + A_{pq} \text{ (potential energy)}$$

$$T_{pq} = -\frac{1}{2} \int d\vec{r} e^{-\alpha_p r^2} \nabla^2 e^{-\alpha_q r^2} = 3 \frac{\alpha_p \alpha_q \pi^{3/2}}{(\alpha_p + \alpha_q)^{5/2}}$$

$$A_{pq} = - \int d\vec{r} e^{-\alpha_p r^2} \frac{1}{r} e^{-\alpha_q r^2} = -\frac{2\pi}{\alpha_p + \alpha_q}$$

- Program: `Hatomv1.f` (FORTRAN implementation) has four Gaussian coefficients ( $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ )

## Exercise 3.4: The finite potential well (I)

$$V(x) = \begin{cases} 0 & \text{for } |x| > a \\ -V_0 & \text{for } |x| \leq a \end{cases}$$

Stationary states in an even potential ( $V(-x) = V(x)$ ). The Hamiltonian commutes with the parity operator. Stationary states are either “odd” or “even”

Even parity:  $\psi_m^E(-x) = +\psi_m^E(x)$   
Odd parity:  $\psi_m^O(-x) = -\psi_m^O(x)$

It is clear that:

$$\langle \psi_m^E | \hat{O} | \psi_n^O \rangle = 0 \text{ for, } \hat{O} \in \{\hat{T}, \hat{V}, \hat{1}\}$$

Accordingly: in order to save on computing time the “even” and “odd” states can be treated separately (parity conservation).

## Exercise 3.4: The finite potential well (II)

Approach 1: Expand the wave functions for the finite potential well in terms of standing waves in  $[-L, +L]$  (periodicity imposed by the basis functions)

$$\psi_n(x) = \frac{1}{\sqrt{2L}} e^{ik_n x} \quad \text{with, } k_n = \frac{n\pi}{2L} \quad \text{and,} \quad n = \pm 1, \pm 2, \pm 3, \dots$$

It is clear that  $L \gg a$ . The overlap matrix  $S$  and  $H = T + A$

$$S_{mn} = \frac{1}{2L} \int_{-L}^{+L} dx e^{i(k_n - k_m)x} = \delta_{nm}$$

$$T_{mn} = \frac{1}{2L} \int_{-L}^{+L} dx e^{-ik_m x} \frac{-\hbar^2}{2m} \frac{d^2}{dx^2} e^{ik_n x} = \frac{\hbar^2 k_n^2}{2m} \delta_{nm}$$

$$\begin{aligned} A_{mn} &= \frac{1}{2L} \int_{-L}^{+L} dx e^{-ik_m x} V(x) e^{ik_n x} \\ &= \frac{-V_0 a}{L} \delta_{nm} - \frac{V_0}{L} \frac{\sin(k_m - k_n) a}{(k_m - k_n)} \end{aligned}$$

## Exercise 3.4: The finite potential well (III)

Approach 2 (Even eigenfunctions): Expand the wave functions for the finite potential well in terms of real part of the standing waves in  $[-L, +L]$ . It is clear that  $L \gg a$ .

$$\psi_n(x) = \frac{1}{\sqrt{L}} \cos(k_n x) \quad \text{with, } k_n = \frac{n\pi}{2L} \quad \text{and,} \quad n = 1, 3, 5, 7 \dots$$

The  $S$  and  $T$  (kinetic energy) overlap matrices adopt the following expression

$$S_{mn} = \frac{1}{L} \int_{-L}^{+L} dx \cos(k_m x) \cos(k_n x) = \delta_{nm}$$

$$T_{mn} = \frac{1}{L} \int_{-L}^{+L} dx \cos(k_m x) \frac{-\hbar^2}{2m} \frac{d^2}{dx^2} \cos(k_n x) = \frac{\hbar^2 k_n^2}{2m} \delta_{nm}$$

## Exercise 3.4: The finite potential well (IV)

Overlap matrix for the potential energy

$$A_{nn} = \frac{1}{L} \int_{-L}^{+L} dx \cos(k_n x) V(x) \cos(k_n x)$$

$$= \frac{-V_0}{L} \left[ a + \frac{\sin(2k_n a)}{(2k_n)} \right]$$

$$A_{mn} = \frac{1}{L} \int_{-L}^{+L} dx \cos(k_m x) V(x) \cos(k_n x)$$

$$= \frac{-V_0}{L} \left[ \frac{\sin(k_m - k_n) a}{(k_m - k_n)} + \frac{\sin(k_m + k_n) a}{(k_m + k_n)} \right] \quad (m \neq n)$$

It is clear that both  $S$  and  $H = T + A$  are symmetric matrices:  
 $S_{mn} = S_{nm}$  and  $H_{mn} = H_{nm}$ .

## Exercise 3.4: The finite potential well (V)

Approach 3 (Odd eigenfunctions): Expand the wave functions for the finite potential well in terms of imaginary part of the standing waves in  $[-L, +L]$ . It is clear that  $L \gg a$ .

$$\psi_n(x) = \frac{1}{\sqrt{L}} \sin(k_n x) \quad \text{with, } k_n = \frac{n\pi}{2L} \quad \text{and,} \quad n = 2, 4, 6, 8 \dots$$

The  $S$  and  $T$  (kinetic energy) overlap matrices adopt the following expression

$$S_{mn} = \frac{1}{L} \int_{-L}^{+L} dx \sin(k_m x) \sin(k_n x) = \delta_{nm}$$

$$T_{mn} = \frac{1}{L} \int_{-L}^{+L} dx \sin(k_m x) \frac{-\hbar^2}{2m} \frac{d^2}{dx^2} \sin(k_n x) = \frac{\hbar^2 k_n^2}{2m} \delta_{nm}$$

## Exercise 3.4: The finite potential well (VI)

Overlap matrix for the potential energy

$$A_{nn} = \frac{1}{L} \int_{-L}^{+L} dx \sin(k_n x) V(x) \sin(k_n x)$$

$$= \frac{-V_0}{L} \left[ a - \frac{\sin(2k_n a)}{(2k_n)} \right]$$

$$A_{mn} = \frac{1}{L} \int_{-L}^{+L} dx \sin(k_m x) V(x) \sin(k_n x)$$

$$= \frac{-V_0}{L} \left[ \frac{\sin(k_m - k_n) a}{(k_m - k_n)} - \frac{\sin(k_m + k_n) a}{(k_m + k_n)} \right] \quad (m \neq n)$$

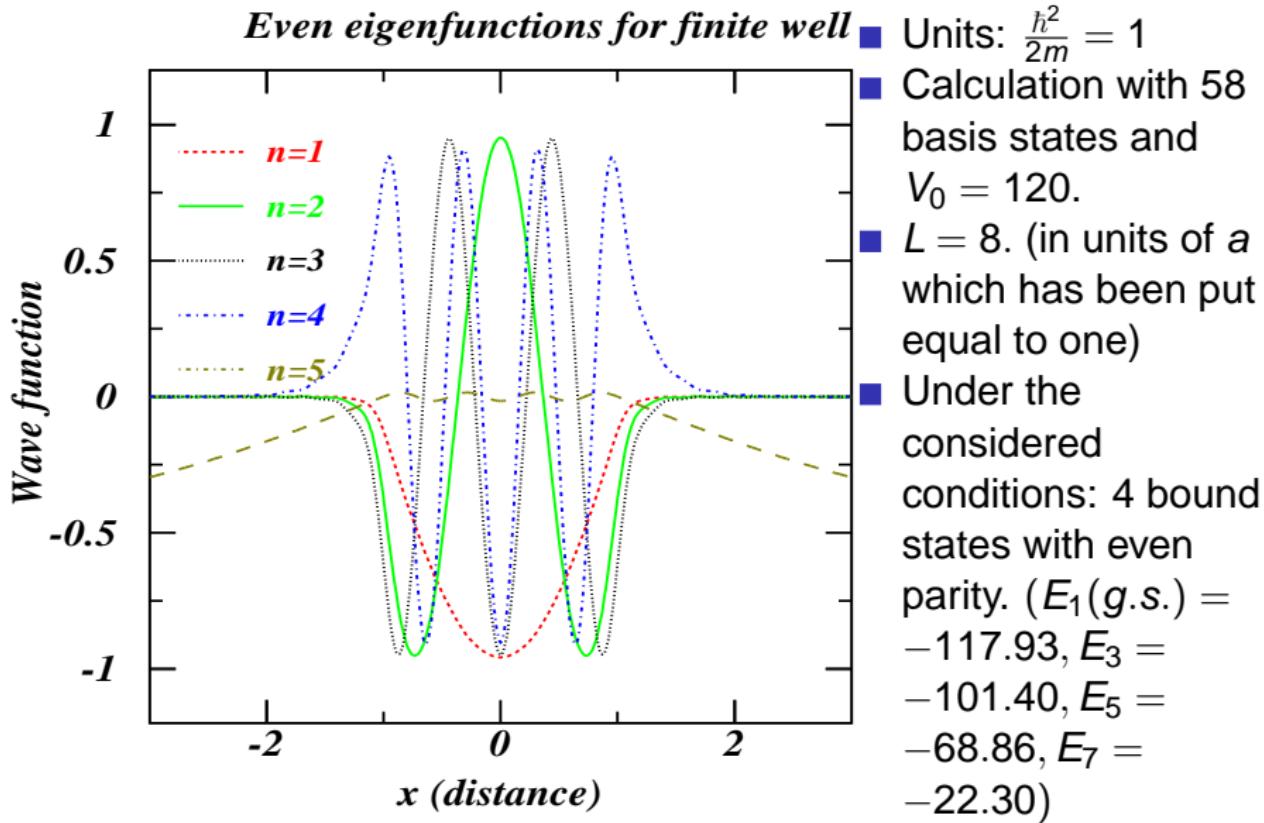
It is clear that both  $S$  and  $H = T + A$  are symmetric matrices:  
 $S_{mn} = S_{nm}$  and  $H_{mn} = H_{nm}$ .

## Exercise 3.4: The finite potential well (VII)

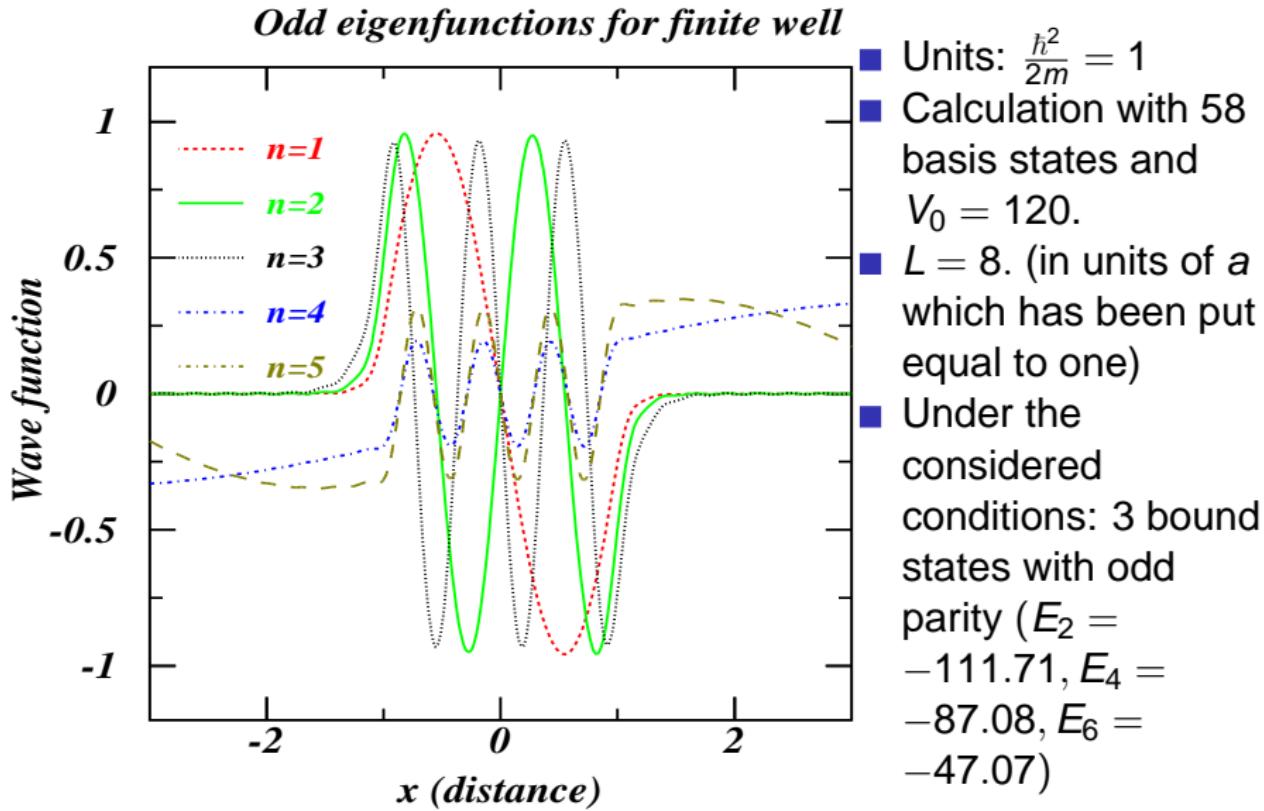
The program FINITEWELL.F adopts the following units

- $a$  is the unit of length ( $a = 1$ )
- $\frac{\hbar^2}{2m} = 1$

# Lowest 5 even eigenfunctions for finite well



# Lowest 5 odd eigenfunctions for finite well



### 3.4 Perturbation theory and variational calculus (I)

- Hilbert space (states  $(p, q)$ ): set A (states  $(n, m)$ ) and set B (states  $(\alpha, \beta)$ ) (ORTHONORMAL STATES)
- Introduce  $H'_{pq}$  without diagonal elements:  $H'_{pq} = H_{pq} (1 - \delta_{pq})$
- Rewrite the eigenvalue equation for orthonormal states  $\sum_{q=1}^{q=N} (E\delta_{pq} - H_{pq}) C_q = 0 (\forall p)$  in terms of  $H'_{pq}$

$$(E - H_{pp}) C_p = \sum_{n \in A} H'_{pn} C_n + \sum_{\alpha \in B} H'_{p\alpha} C_\alpha$$

- Introduce  $h'_{pn} \equiv \frac{H'_{pn}}{(E - H_{pp})}$  and  $h'_{p\alpha} \equiv \frac{H'_{p\alpha}}{(E - H_{pp})}$  so that the above can be rewritten

$$C_p = \sum_{n \in A} h'_{pn} C_n + \sum_{\alpha \in B} h'_{p\alpha} C_\alpha \quad (\forall p)$$

- Start the iteration with the  $C_\alpha$ :  $C_\alpha = \sum_{n \in A} h'_{\alpha n} C_n + \sum_{\beta \in B} h'_{\alpha\beta} C_\beta$

### 3.4 Perturbation theory and variational calculus (II)

- After the first iteration

$$C_p = \sum_{n \in A} h'_{pn} C_n + \sum_{\alpha \in B} h'_{p\alpha} \left[ \sum_{n \in A} h'_{\alpha n} C_n + \sum_{\beta \in B} h'_{\alpha\beta} C_\beta \right]$$

- Repeat the procedure of eliminating the  $C_\beta$  to obtain

$$C_p = \sum_{n \in A} \left[ h'_{pn} + \sum_{\alpha \in B} h'_{p\alpha} h'_{\alpha n} + \sum_{\alpha, \beta \in B} h'_{p\alpha} h'_{\alpha\beta} h'_{\beta n} + \dots \right] C_n$$

- Introduce the notation

$$U_{pn}^A \equiv H_{pn} + \sum_{\alpha \in B} \frac{H'_{p\alpha} H'_{\alpha n}}{(E - H_{\alpha\alpha})} + \sum_{\alpha, \beta \in B} \frac{H'_{p\alpha} H'_{\alpha\beta} H'_{\beta n}}{(E - H_{\alpha\alpha})(E - H_{\beta\beta})} + \dots$$

To obtain our final expression:

$$C_p = \sum_{n \in A} \frac{(U_{pn}^A - H_{pn}\delta_{pn})}{(E - H_{pp})} C_n$$

### 3.4 Perturbation theory and variational calculus (III)

- Per-Olov Löwdin: Honorary Doctoral Degree from Ghent University (1975)
- For  $p \in A$  ( $p \equiv m$ ) we obtain the eigenvalue equation

$$EC_m = \sum_{n \in A} U_{mn}^A C_n$$

**Löwdin method (1956): extra basis states in  $B$  are taken into account in a perturbative manner, the dimension is determined by the amount of basis states in  $A$**

- The above equation is difficult to solve
  - 1  $U_{mn}^A$  contains the energy that remains to be calculated
  - 2 in practice: fixed  $E$  is chosen somewhere in the region for which accurate results are required

## 3.4 Perturbation theory and variational calculus (IV)

- Löwdin method is frequently used for electrons in solids (for electrons in solids one can take  $E \approx E_F$  as those are the electrons for which we wish accurate results)
- the convergence of  $U_{pn}^A$  depends on the magnitude of the matrix elements  $H'_{p\alpha}$  and  $H'_{\alpha\beta}$  (e.g. convergence can be guaranteed in a plane wave basis if potential varies substantially slower than the wavelengths in set B )

# Chapter 7: Classical statistical mechanics

- 7.1 Basic theory
  - 7.1.1 Ensembles
- 7.2 Examples of statistical models - phase transitions
  - 7.2.1 Molecular systems
  - 7.2.2 Lattice models (**Ising model**)
- 7.3 Phase transitions (**NIET**)
  - 7.3.1 First order and continuous phase transitions
  - 7.3.2 Critical phase transitions and finite size scaling
- 7.4 Determination of averages in simulations

Extra course notes: Chapter 7 of “Statistical Physics I” (J.R.)

## 7.1 Basic theory - Introduction I

- **Statistical mechanics** concerns the study of systems with many degrees of freedom (dof).
  - 1  $(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots, \vec{r}_N, \vec{p}_1, \vec{p}_2, \vec{p}_3, \dots, \vec{p}_N)$  with  $N$  large.
  - 2 Magnetic moments (or, spins)
- **Phase space:** space spanned by the dof. Every point in phase space represents a particular configuration of the system.
- In the course of time: system follows a path in phase space
- Usually one is only interested in the physical quantities, which are related to the time averages of these dof. Indeed, our measuring devices (thermometers, barometers) are “slow” on an atomic scale!
- Law of large numbers (for the uncorrelated contributions!) makes fluctuations about the time average very small (fluctuations  $\sim 1/\sqrt{N}$  even in correlated systems)

## 7.1 Basic theory - Introduction II

- **Computer simulations sample relatively few dof !**
- Results are representative if
  - 1 spatial correlations of the physical system extend over range smaller than the system size in the simulation
  - 2 correlation time in the physical system smaller than the simulation time
- Finite-size scaling: extract useful information from simulations with a system smaller than the correlation length by extrapolation.
- Time average of a physical quantity

$$\bar{A} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T A(t) dt$$

- In molecular dynamics (Chapter 8): typical simulation time  $10^{-9} \text{ s} \leq T \leq 10^{-6} \text{ s}$ , far below the time in which most measuring devices sample physical quantities !

## 7.1.1 Basic theory - Ensembles I



- System parameters: physical quantities which are kept “fixed” or whose average values is controlled externally
- Adapting the simulation techniques for classical many-particle systems (*molecular dynamics* and *Monte-Carlo*) to these experimental situations is a non-trivial problem and requires profound insight into the principles of statistical mechanics!
- Ensemble theory is of great use for simulation in order to mimic the operation of the system parameters

## 7.1.1 Basic theory - Ensembles II

Fundamental postulates of statistical mechanics pertain to systems with  $(E, V, N)$  system parameters!

*(Magnetic systems:  $(E, H, N)$ , with  $H$  external magnetic field)*

- **Ergodic hypothesis:** all  $(E, V, N)$  states accessible to the system are equally likely to be visited in the course of time! As a result, the time average can be associated with an average over all possible states!
- **Ensemble average:**

$$\bar{A} \equiv \langle A \rangle = \frac{\sum_{X|E} A(X)}{\sum_{X|E}} = \frac{\sum_X A(X) \delta(\mathcal{H}(X) - E)}{\sum_X \delta(\mathcal{H}(X) - E)}$$

$\sum_X$ : Sum over phase space  $X$  at constant  $(E, V, N)$ ! Set of states under consideration :  **$(E, V, N)$  or Microcanonical Ensemble**

- In the “classical” situation (theory of classical gases, liquids, solids)

$$\sum_X \rightarrow \frac{1}{h^{3N}} \int_V d\vec{r}_1 d\vec{r}_2 d\vec{r}_3 \dots d\vec{r}_N \int d\vec{p}_1 d\vec{p}_2 d\vec{p}_3 \dots d\vec{p}_N$$

- Number of states with energy  $E$  (in quantum mechanics particles are indistinguishable!)

$$\Omega(N, V, E) = \frac{1}{N!} \sum_X \delta(\mathcal{H}(X) - E)$$

States which can be obtained by permuting particles should be counted only once !

- **Entropy  $S(N, V, E)$**   $S(N, V, E) = k_B \ln \Omega(N, V, E)$

$$T = \left( \frac{\partial S}{\partial E} \right)_{N,V}^{-1} \quad \mu = -T \left( \frac{\partial S}{\partial N} \right)_{E,V} \quad P = T \left( \frac{\partial S}{\partial V} \right)_{N,E}$$

Experimental situations: often temperature  $T$  is kept constant!

$(N, V, T)$  or canonical ensemble

$$\langle A \rangle_{NVT} = \frac{1}{N!} \frac{\sum_X A(X) \exp -\beta \mathcal{H}(X)}{Z(N, V, T)}$$

Partition function

$$Z(N, V, T) = \frac{1}{N!} \sum_X \exp -\beta \mathcal{H}(X) = \sum_E \Omega(N, V, E) \exp -\beta E$$

Free Energy  $F(T, V, N)$   $F = E - TS = -k_B T \ln Z(N, V, T)$

$$\mu = \left( \frac{\partial F}{\partial N} \right)_{T, V} \quad P = - \left( \frac{\partial F}{\partial V} \right)_{N, T} \quad S = - \left( \frac{\partial F}{\partial T} \right)_{V, N}$$

$$Z(N, V, T) = \sum_E \exp -\beta(E - TS(E)) = \sum_E \exp -\beta F_E(T, V, N)$$

$Z(N, V, T)$  dominated by the states for which  $F_E$  is minimal!

## ( $N, P, T$ ) or constant pressure ensemble

$$\langle A \rangle_{NPT} = \frac{1}{N!} \frac{\int dV \exp -\beta PV \sum_X A(X) \exp -\beta \mathcal{H}(X)}{Q(N, P, T)}$$

## Partition function

$$\begin{aligned} Q(N, P, T) &= \int dV e^{-\beta PV} \frac{1}{N!} \sum_X \exp -\beta \mathcal{H}(X) \\ &= \int dV [Z(N, V, T)] \exp -\beta PV \end{aligned}$$

## Gibbs Free Energy of Gibbs potential $G(T, P, N)$

$$G = E + PV - TS = -k_B T \ln Q(N, P, T)$$

$$\mu = \left( \frac{\partial G}{\partial N} \right)_{T, P} \quad V = \left( \frac{\partial G}{\partial P} \right)_{N, T} \quad S = - \left( \frac{\partial G}{\partial T} \right)_{P, N}$$

## $(\mu, V, T)$ or grand-canonical ensemble

$$\langle A \rangle_{\mu VT} = \sum_N \exp \beta \mu N \frac{1}{N!} \frac{\sum_X A(X) \exp -\beta \mathcal{H}(X)}{Z_G(\mu, V, T)}$$

## Grand Canonical partition function

$$Z_G(\mu, V, T) = \sum_N Z(N, V, T) \exp \beta \mu N$$

## Grand canonical potential $\Omega_G(\mu, V, T)$

$$\Omega_G(\mu, V, T) = -k_B T \ln Z_G(\mu, V, T) = F - \mu N = E - TS - \mu N$$

$$N = \left( \frac{\partial \Omega_G}{\partial \mu} \right)_{T, V} \quad P = - \left( \frac{\partial \Omega_G}{\partial V} \right)_{\mu, T} \quad S = - \left( \frac{\partial \Omega_G}{\partial T} \right)_{V, \mu}$$

# Expectation values of thermodynamic quantities I

Expectation values are either computed as **ENSEMBLE AVERAGES** or as **INTEGRALS OVER PHASE SPACE**.

- Example of an ensemble average: compute the internal energy of the system in the canonical ensemble

$$\langle E \rangle_{NVT} = \frac{\sum_X \mathcal{H}(X) \exp -\beta \mathcal{H}(X)}{\sum_X \exp -\beta \mathcal{H}(X)} = -\frac{\partial \ln Z(N, V, T)}{\partial \beta}$$

Or, compute specific heat at constant volume  $C_V = \left(\frac{\partial E}{\partial T}\right)_{NV}$

$$C_V = \frac{1}{k_B T^2} \frac{\partial^2 \ln Z(N, V, T)}{\partial \beta^2} = -\frac{1}{k_B T^2} \left[ \left\langle E^2 \right\rangle_{NVT} - \langle E \rangle_{NVT}^2 \right]^2$$

- Example of an integral over phase space: the pressure

$$\frac{PV}{Nk_B T} = 1 - \frac{1}{3Nk_B T} \left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{\nabla}_i V(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) \right\rangle$$

# Expectation values of thermodynamic quantities II

- Molecular dynamics and Monte-Carlo method: one can calculate ensemble averages of physical quantities  $\hat{O}$  which can be expressed in terms of the system coordinates (e.g.  $\hat{O} = O(\vec{r}_i, \vec{p}_i)$ ):  
**MECHANICAL AVERAGES**
- Unlike  $P$ :  $F$  and  $\mu$  cannot be calculated as mechanical averages (=integrations over physical coordinates)
- Example of alternative method: Given  $F$  at  $T$  how to compute it at  $T'$ ? Differences can be expressed *as ensemble averages*

$$\begin{aligned}\exp[\beta F(T, V, N) - \beta' F(T', V, N)] &= \frac{Z(N, V, T')}{Z(N, V, T)} \\ &= \frac{\sum_X \exp [(-\beta' + \beta) \mathcal{H}(X)] \exp -\beta \mathcal{H}(X)}{Z(N, V, T)} \\ &= \langle \exp [(-\beta' + \beta) \mathcal{H}] \rangle_{\beta}\end{aligned}$$

# Expectation values of thermodynamic quantities III

- Disadvantage of this method to determine  $F(N, V, T')$ : bad statistics unless the temperatures  $T$  and  $T'$  are sufficiently close!  
*(indeed simulation system at temperature  $T$  will seek those configurations with minimal free energy at  $T$ )*
- **Thermodynamic integrations**

$$F(T, V_1, N) = F(T, V_0, N) - \int_{V_0}^{V_1} P(T, V, N) dV$$

Requires that one performs simulations to determine  $\langle P(T, V, N) \rangle$  as a function of  $V$  for fixed values of  $(T, N)$ .

- If one can perform simulations to determine  $\langle E(T, V, N) \rangle$  as a function of  $T$  for fixed values of  $(V, N)$ .

$$\frac{F(T_1, V)}{T_1} = \frac{F(T_0, V)}{T_0} - \int_{T_0}^{T_1} \frac{E(T, V)}{T^2} dT$$

## 7.2.1 Examples of statistical models: molecular systems

### Determining a model

- 1 Requires a definition of the dof.
- 2 Requires a Hamiltonian  $\mathcal{H}$ .

Molecular models :  $N$  identical spherically symmetric particles with the following Hamiltonian

$$\mathcal{H}(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N, \vec{p}_1, \vec{p}_2, \dots, \vec{p}_N) = \sum_{i=1}^N \frac{p_i^2}{2m} + V_N(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$$

with,  $V_N(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) = \frac{1}{2} \sum_{i \neq j=1}^N V_2(|\vec{r}_i - \vec{r}_j|)$ .

- pair potentials are popular
- determining potential energy is time consuming ( $\sum_{i < j} \sim N^2$ )
- three-body forces are VERY expensive! ( $\sum_{i < j < k} \sim N^3$ )

## Canonical partition function of classical molecular system

$$Z(N, V, T) = \frac{1}{h^{3N} N!} \int d\vec{r}_1 d\vec{r}_2 \dots d\vec{r}_N \int d\vec{p}_1 d\vec{p}_2 \dots d\vec{p}_N \\ \times \exp \left\{ -\beta \left( \sum_{i=1}^N \frac{p_i^2}{2m} + V_N(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) \right) \right\}$$

Integration over momenta coordinates is straightforward !

$$Z(N, V, T) = \frac{1}{N!} \left( \frac{2\pi m k T}{h^2} \right)^{\frac{3N}{2}} \int d\vec{r}_1 d\vec{r}_2 \dots d\vec{r}_N \\ \times \exp -\beta V_N(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$$

- 1 in systems with rigid polyatomic molecules: rigidity constraints and atomic pair potentials
- 2 tantalising problem: simulation of water using ab initio interaction potentials

# Molecular systems: determine macroscopic quantities

Determine the macroscopic quantities in the simulation and compare results with measurements

- The pressure can be determined from the virial theorem

$$\frac{PV}{Nk_B T} = 1 - \frac{1}{3Nk_B T} \left\langle \sum_{i=1}^{i=N} \vec{r}_i \cdot \vec{\nabla}_i V(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) \right\rangle$$

- the specific heat  $C_V \sim (\Delta E)^2$  but  $E$  is constant in canonical ensemble
- $C_V$  can be determined from fluctuations in the kinetic energy  $K$

$$\frac{\Delta K^2}{\langle K \rangle^2} = \frac{2}{3N} \left( 1 - \frac{3N}{2C_V} \right)$$

- a lot of quantities can be extracted from the knowledge of the static pair correlation function (radial distribution function)

# Static pair correlation function

Link between experimental observables and simulations is often established by means of correlation functions.

## Static pair correlation function

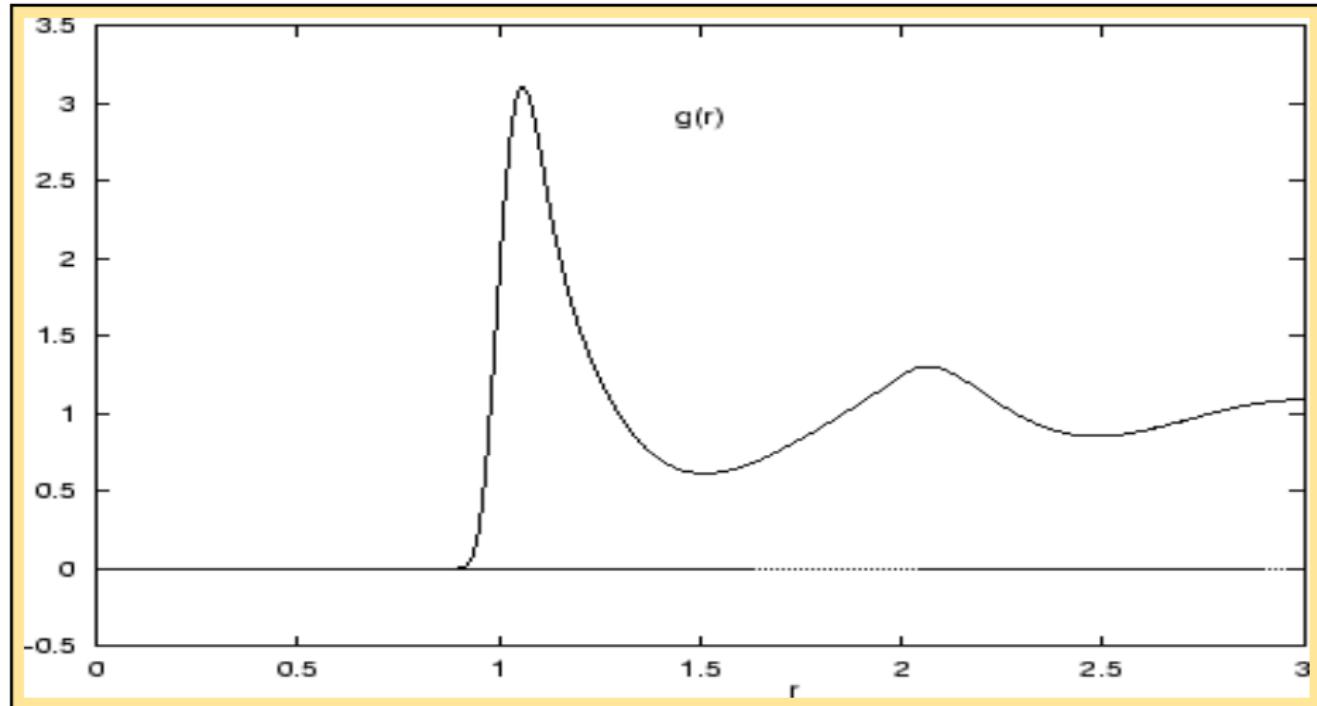
$$g(\vec{r}, \vec{r}') = \frac{V^2}{N(N-1)} \left\langle \sum_{i \neq j}^N \delta(\vec{r} - \vec{r}_i) \delta(\vec{r}' - \vec{r}_j) \right\rangle$$

Proportional to probability of finding a particle at the position  $\vec{r}$  and simultaneously one at  $\vec{r}'$ . For a homogeneous system

$$g(\Delta \vec{r}) = \frac{V}{N(N-1)} \left\langle \int d\vec{r}' \sum_{i \neq j}^N \delta(\vec{r}' - \vec{r}_i) \delta(\vec{r}' + \Delta \vec{r} - \vec{r}_j) \right\rangle$$

- 1 With  $g(\Delta \vec{r})$  one can compute the pressure  $P$
- 2 For large  $|\Delta \vec{r}|$ : correlation disappears and  $g(\Delta \vec{r}) \rightarrow 1$ .
- 3 Bare correlation function:  $h(\Delta \vec{r}) = g(\Delta \vec{r}) - 1$

The pair correlation function contains information about the local structure of the fluid (*long-distance behaviour is homogeneous, but short-range structure is inhomogeneous*)



Can be experimentally determined through X-ray scattering!

# Velocity autocorrelation function I

Another important correlation function: VAF or velocity autocorrelation function (*correlations in time!*)

- Definition:

$$c_{v_i}(t) = \langle \vec{v}_i(t=0) \cdot \vec{v}_i(t) \rangle .$$

Physical interpretation: dot product of a tagged (but arbitrary) particle at time 0 with the velocity of the same particle at time  $t$

- For a homogeneous system independent of  $i$ .

$$c_v(t) = \frac{1}{N} \lim_{T \rightarrow \infty} \sum_{i=1}^N \frac{1}{T} \int_0^T dt' \langle \vec{v}_i(t') \cdot \vec{v}_i(t' + t) \rangle .$$

Combined average over time and particles!

- The  $c_v(t)$  is a dynamic quantity and cannot be found as an ensemble average. Ensemble averages are suitable for evaluation of averages of static quantities!

## Velocity autocorrelation function II

- Simulations have taught us that  $c_v(t)$  does NOT decay exponentially! In  $D$  dimensions

$$c_v(t) \sim \frac{1}{t^{\frac{D}{2}}}$$

*Particle moving in a fluid does not easily forget its initial motion! It turns out that the tagged particle causes a pressure rise ahead and a pressure drop behind itself and the resulting pressure differences produce vortices (in two dimensions) and these persist over relatively long time scales.*

*B.J. Alder en T.W. Wainwright, Phys. Rev. A1 (1970) 18-21*

- “Molecular chaos” assumption (VAF decays exponentially) is NOT always valid
- Remarkable quantitative agreement with hydrodynamic calculations for a sphere moving in a liquid!  
*(duality between microscopic and macroscopic approaches!)*

## 7.4 Determination of averages in simulations I

**How to determine the physical quantities and how large are the errors  $\epsilon$  on the computed averages?**

- In MD and MC simulations physical quantities are determined as averages (or, “time averages”) over the  $M$  configurations generated in the simulation

$$\bar{A} = \frac{1}{M} \sum_{N=1}^M A_n$$

- If the system size of the simulation and simulation time are large enough, these averages will be very close to the averages in a macroscopic experimental system
- Standard deviation of the physical quantity  $A$  ( $\langle \dots \rangle$  denotes ensemble average over independent simulations)

$$\sigma^2 = \langle A^2 \rangle - \langle A \rangle^2$$

Requires simulations with independent configurations!

## 7.4 Determination of averages in simulations II

- The standard deviation can be ESTIMATED by computing the time average  $\sigma^2 \approx \overline{A^2} - \overline{A}^2$  over the  $M$  configurations created in the simulation
- Problem: not all configurations generated in the simulation can be made INDEPENDENT! There is some correlation time  $\tau$  (time correlations between subsequent samples generated by the MD or MC simulation).
- The time correlations in the simulation manifest themselves in the time correlation function
- The standard deviation of the mean value of the  $A$  calculated over  $M$  samples generated in the simulation should be determined in terms of the independent samples generated in the simulation (*total number of simulation samples divided by the correlation time measured in simulation steps*)

## 7.4 Determination of averages in simulations III

### Time correlation function

(How independent are consecutive simulations?)

$$\begin{aligned} c_{AA}(k) &= \left\langle (A_n - \langle A_n \rangle)(A_{n+k} - \langle A_{n+k} \rangle) \right\rangle \\ &= \left\langle A_n A_{n+k} \right\rangle - \left\langle A_n \right\rangle \left\langle A_{n+k} \right\rangle \end{aligned}$$

$\langle A_n \rangle$ : average of  $A$  at time instance  $n$

- 1  $c_{AA}(k)$  independent of  $n$  because of time translation symmetry
- 2 for  $k = 0$ :  $c_{AA}(k = 0) = \sigma^2$
- 3 time correlations over  $k$  manifest themselves as  $c_{AA}(k) \neq 0$  for  $k = 1, 2, 3, \dots$
- 4 absence of time correlations over  $k$  reflects itself in  $c_{AA}(k)$  assuming zero value

## 7.4 Determination of averages in simulations IV

### ■ Correlation time $\tau$

$$\tau = \frac{\int_0^{+\infty} t [A(t) - \bar{A}] dt}{\int_0^{+\infty} [A(t) - \bar{A}] dt}$$

### ■ Integrated correlation time $\tau_{int}$

$$\tau_{int} = \frac{1}{2} \sum_{n=-\infty}^{n=+\infty} \frac{c_{AA}(n)}{c_{AA}(0)}$$

### ■ Exponential correlation time $\tau_{exp}$

$$\tau_{exp} = \frac{-t}{\ln \left| \frac{c_{AA}(t)}{c_{AA}(0)} \right|} \quad \text{for sufficiently large } t$$

## 7.4 Determination of averages in simulations IV

- **Correlation time:** Time over which a physical quantity  $A$  relaxes towards its equilibrium time (relaxation time!). Only useful at the beginning of a simulation with realistic dynamics.
- **Integrated correlation time:** Can be used throughout the simulation and not only at the beginning when the quantity  $A$  decays to its equilibrium value.
- **Exponential correlation time:**  $\tau_{\text{exp}}$  is the slowest decay time with which the system relaxes to its equilibrium (such as happens at the start of a simulation when the system is not yet in equilibrium). In general, not equal to the integrated correlation time.

## 7.4 Determination of averages in simulations V

How to determine the standard deviation to  $\bar{A}$ ? Standard deviation  $\epsilon$  (or, “estimated error”) of the mean value of the quantity  $A$  in a simulation with  $M$  configurations.

- Say that we have  $M$  subsequent values of  $A$  in a simulation
- In the absence of time correlations in the simulation

$$\sigma^2 = \bar{A^2} - (\bar{A})^2 = \frac{1}{M} \sum_{n=1}^M A_n^2 - \left( \frac{1}{M} \sum_{n=1}^M A_n \right)^2$$

determines  $\epsilon^2$  by means of

$$\epsilon^2 = \frac{\sigma^2}{M} .$$

## 7.4 Determination of averages in simulations VI

- When time correlations occur the standard deviation in the mean is given by

$$\begin{aligned}\epsilon^2 &= \frac{1}{M^2} \sum_{n,m=1}^M \langle A_n A_m \rangle - \left( \frac{1}{M} \sum_{n=1}^M \langle A_n \rangle \right)^2 \\ &= \sum_{n,m=1}^M \frac{1}{M^2} (\langle A_n A_m \rangle - \langle A_n \rangle \langle A_m \rangle) \\ &= \frac{1}{M^2} \sum_{n,m=1}^M c_{AA}(n-m) = \frac{1}{M^2} \sum_{n=1}^M \sum_{l=1-n}^{M-n} c_{AA}(l)\end{aligned}$$

- 1 lowest possible value for  $l$ :  $-(M-1)$
- 2 highest possible value for  $l$ :  $(M-1)$
- 3 each value of  $l$  occurs  $(M - |l|)$  times

## 7.4 Determination of averages in simulations VII

- For  $M$  uncorrelated samplings of a Gaussian distributed variable the error is given by  $\frac{\sigma}{\sqrt{M}}$
- Standard deviation in the mean in a simulation which involves correlations

$$\epsilon^2 = \frac{1}{M} \sum_{I=-(M-1)}^{M-1} \left(1 - \frac{|I|}{M}\right) c_{AA}(I) \xrightarrow{\text{large } M} 2 \frac{\tau_{int}}{M} c_{AA}(0) = 2 \frac{\tau_{int}}{M} \sigma^2$$

- **Time correlations cause the error to be multiplied by  $\sqrt{2\tau_{int}}$ !**
- STATISTICAL ERROR  $\epsilon$  on the computed average of a variable can be determined from the STANDARD DEVIATION  $\sigma$  and the INTEGRATED CORRELATION TIME  $\tau_{int}$
- Computing the  $\tau_{int}$  is often a tedious task which requires several series of independent simulations with  $M$  configurations!

## 7.4 Determination of averages in simulations VIII

In practice: simpler method ( “**data-blocking**”)

- 1 The results of the physical quantities in the  $M$  simulations are recorded in a file ( $A_i, i = 1, 2, \dots, M$ )
- 2 The data sequence is chopped into blocks of size  $m$  which is larger than the correlation time
- 3 For blocks of size  $m$ , the  $j$ -th block average is given by

$$\bar{A}_j = \frac{1}{m} \sum_{k=jm+1}^{k=jm+j} A_k$$

- 4 The averages  $\bar{A}_j$  in the different blocks are uncorrelated!
- 5 Error: standard deviation of the uncorrelated block averages
- 6 Check: errors should be independent of the block size (if block size is sufficiently larger than the correlation time!)

## 7.4 Determination of averages in simulations IX

Quantity A: Gaussian distributed with mean  $\langle A \rangle$  and width  $\sigma$

- Estimator for the mean of this distribution (requires  $M$  statistically independent observations  $\{A_i\}$ )

$$\bar{A} = \frac{1}{M} \sum_{i=1}^{i=M} A_i$$

and standard error for this estimate:  $(\text{error})^2 = \frac{\sigma^2}{M}$

- Most simulation techniques lead to “dynamic” correlations between subsequently generated observations  $\{A_i\}$ . The standard error becomes

$$(\text{error})^2 = \frac{\sigma^2}{M} \left( 1 + \frac{2\tau_{int}}{\delta t} \right)$$

where  $\delta t$  is the time interval between subsequently generated states  $A_i, A_{i+1}$  and  $\tau_{int}$  is the integrated correlation time (expressed in units  $\delta t$ )!

# Chapter 8 : Molecular dynamics simulations

- Appendix B: Random Number Generators
- 8.1 Introduction
- 8.2 Molecular dynamics at constant energy
- 8.3 A molecular dynamics simulation program for argon
- 8.4 Integration methods - symplectic integrators
  - 8.4.1 The Verlet algorithm revisited
  - 8.4.2 Symplectic geometry ; symplectic integrators (NIET)
  - 8.4.3 Derivation of symplectic integrators (NIET)
- 8.5 Molecular dynamics methods for different ensembles
  - 8.5.1 Constant temperature
  - 8.5.2 Keeping the pressure constant
- 8.6 Molecular systems (NIET)
- 8.7 Long range interactions (NIET)
- 8.8 Langevin dynamics simulation
- 8.9 Dynamical quantities - nonequilibrium molecular dynamics

## 8.1 Introduction

- Impossible to generate random sample of possible system configurations! (there are simply too many of these)
- With Molecular Dynamics (MD) and Monte-Carlo (MC) technique: generate a sample of configurations successively
- As a consequence: there are finite time correlations in the MD and MC simulations
- MD integrates the equations-of-motion for a classical many-particle system : simulation of the system as it develops in time! (MD: system moves in phase space along its physical trajectory determined by the equation-of-motion)
- MD allows to study static and dynamical quantities (relaxation and non-equilibrium phenomena)
- MC: system follows a (directed) random walk in phase space
- MC allows to study static quantities and does NOT allow one to study dynamic properties of the system

# General principles of molecular dynamics method

Molecular dynamics is the simulation technique which solves

$$\frac{d^2\vec{r}_i(t)}{dt^2} = \frac{\vec{F}_i(\vec{r})}{m_i} = \frac{\sum_{j=1}^N (j \neq i) F(|\vec{r}_i - \vec{r}_j|) \hat{r}_{ij}}{m_i} \quad (i = 1, \dots, N)$$

for a large collection of  $N$  particles ! Method is approximate

- A1: quantum effects are neglected (quantum effects become important for light particles and/or low temperatures and/or high densities)
- A2: often, forces between particles are not known exactly

- 1 require quantum mechanical calculations ... prone to errors and approximations
- 2 can be obtained by fitting simulations to data
- 3 EXCEPTION: systems of stars and galaxies (large mutual distances and non-relativistic) with  $1/r^2$  Newtonian force

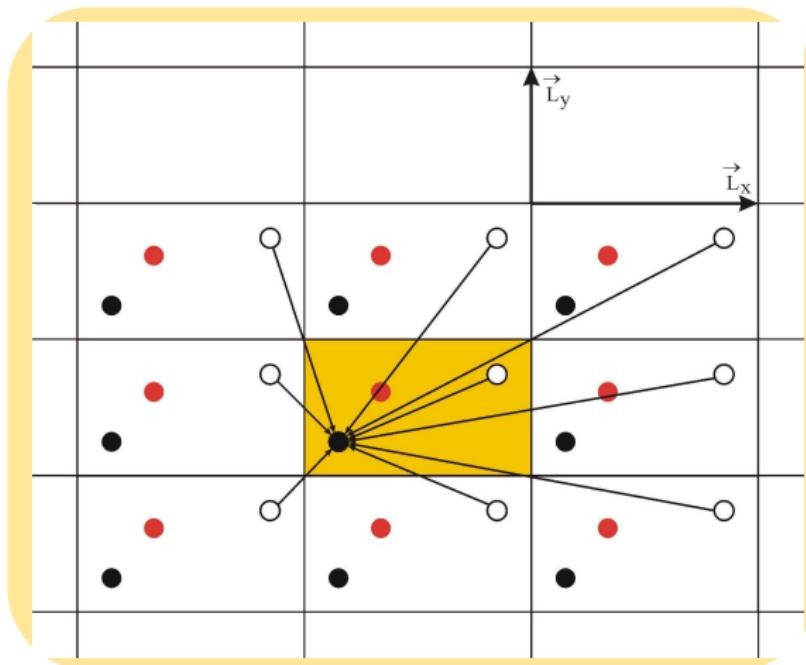
A3: **system sizes** in simulations are much smaller than those of real systems. Reliable simulations for system sizes that are much smaller than correlation length, or, correlation length much larger than system size (finite size scaling method)!

**Periodic Boundary Condition** (PBC): simulate behaviour of a system of a similar size embedded in an infinite system!  
Interaction between particle " $i$ " and " $j$ "

$$\vec{F}(|\vec{r}_i - \vec{r}_j|) \rightarrow \vec{F}_{PBC}(|\vec{r}_i - \vec{r}_j|) = \sum_{\vec{n}(n_1, n_2, n_3)} \vec{F} \left( \left| \vec{r}_i - \left[ \vec{r}_j + \sum_{\mu=1}^3 \vec{L}_\mu n_\mu \right] \right| \right)$$

- 1  $n_\mu$  an integer number, and  $\vec{L}_\mu$  vectors along the edges of the rectangular system.
- 2 infinite sum which requires techniques for approximating it!!!

# Periodic boundary conditions (PBC) I



- standard simulation sizes do have too much surface and insufficient volume (e.g.  $10^3$  particles in 3 dimensions, surface =  $10^3 - 8^3 = 488$  particles)
- with PBC: behaviour of the simulation system is identical to that of real system embedded in the infinite system

#### A4: Time average must be calculated over a finite time!

- 1 Typical time step in the numerical integration of equations-of-motion :  $10^{-14}$  s.
- 2 Correlation time must be smaller than total simulation time.
- 3 Extending the number of time steps does not always help: limitations due to the finiteness of the system size.
- 4 **Recurrence or Poincaré time:** the time after which the system returns to its original configuration!

#### A5: limitations in the numerical integration algorithm!

- 1 optimum choice between speed and accuracy
- 2 time integration step must be carefully chosen

## 8.2 Molecular dynamics at constant energy

## 8.3 A molecular dynamics simulation program for argon

Simulation in an  $(N, V, E)$  ensemble! Three major steps in an MD simulation

- 1 STEP 1: Initialise the MD simulation
- 2 STEP 2: Start the MD simulation and let the system reach equilibrium
- 3 STEP 3: Continue the MD simulation and determine physical quantities

**Step 1: Call to SUBROUTINE Initialise**

**Step 2: Call to SUBROUTINE Simulation (Scale, InitTime)**

**Step 3: Call to SUBROUTINE Simulation (Scale, SimTime)**

**Step 3: Call to SUBROUTINE FinalWrite**

# STEP 1: Initialise the MD simulation (I)

- INPUT: specify the interaction between the particles (Lennard-Jones)
- INPUT with respect to time
  - 1 specify the physical time step between two consecutive system configurations (=`TimeStep`)
  - 2 total time of Step 2 is time needed to let system reach equilibrium =`InitTime`)
  - 3 total time of Step 3 is simulation time =`SimTime`)
- INPUT: number of particles  $N$  (`PartNum`), temperature (`Temperature`), density (`Dichtheid`)

In practice:

***SUBROUTINE InitParameters***

# STEP 1: Initialise the MD simulation (II)

## ■ initialise the positions of the particles

- fcc lattice is the ground state ( $T = 0$ ) for noble gases
- ground state of Lennard-Jones system is hexagonal close-packed
- in fcc lattice: 4 molecules per cel ( $8 \times \frac{1}{8} + 6 \times \frac{1}{2}$ )
- in three dimensions: simulations are often done with  $4M^3$ ,  $M = 1, 2, 3, 4, 5, 6, \dots$  particles
- initial positions: fcc lattice (slightly displaced)
- lattice constant is determined by  $a = L/M$

## ■ initialise the momenta of the particles

: random numbers from a Maxwellian distribution and make sure that the total momentum equals zero  $\left| \sum_{i=1}^N \vec{p}_i \right| = 0$

In practice:

**SUBROUTINE** *InitPositions*  
**SUBROUTINE** *InitMomenta*

## STEP 2: Start simulation and let system reach equilibrium (I)

- Equations of motion are integrated with simple and reliable method: Verlet algorithm. Algorithm succeeds in reasonably conserving the energy during the simulation.

### ***SUBROUTINE Integrate***

- Calculation of forces requires  $\mathcal{O}(N^2)$  steps! Each particle  $i$  interacts with all partners  $j$  in the cell and all copies of  $j$  in the other cells!
- For forces with a sufficiently short range (i.e. not extending about half of the system size):

*Minimum Image Convention*: only the interaction with the nearest copy of each of the remaining particles is retained.

### ***SUBROUTINE CalcPairList(Scale)***

### ***SUBROUTINE UpdatePairList***

# Verlet algorithm (I)

## Orginal Verlet algorithm

- the positions are determined from

$$\vec{r}(t+h) = 2\vec{r}(t) - \vec{r}(t-h) + \frac{h^2}{m} \vec{F}[\vec{r}(t)] + \mathcal{O}(h^4)$$

After a number of integrations the rounding error on  $\vec{r}(t+h)$  becomes  $\mathcal{O}(h^2)$

- the velocities are determined from

$$\vec{v}(t) = \frac{\vec{r}(t+h) - \vec{r}(t-h)}{2h} + \mathcal{O}(h^2)$$

Rounding error is  $h^2$ : higher than the error in the position

- Orginal Verlet algorithm: excellent if one only needs information about the positions!

## Velocity Verlet algorithm

- change of the velocity between  $t$  and  $t + h$

$$\vec{v}(t + h) = \vec{v}(t) + \frac{h}{m} \vec{F}[\vec{r}(t)] + \mathcal{O}(h^2)$$

- better accuracy can be reached through the replacement of the acceleration  $\frac{\vec{F}}{m}$  by the average

$$\frac{1}{m} \left[ \vec{F}[\vec{r}(t)] \right] \longrightarrow \frac{1}{2m} \left[ \vec{F}[\vec{r}(t + h)] + \vec{F}[\vec{r}(t)] \right]$$

- the new position can be obtained by treating the motion between  $t$  and  $t + h$  as one with constant acceleration

$$\vec{r}(t + h) = \vec{r}(t) + h\vec{v}(t) + \frac{h^2}{2m} \vec{F}[\vec{r}(t)] + \mathcal{O}(h^3)$$

## Velocity Verlet algorithm

1 Positions are determined from

$$\vec{r}(t+h) = \vec{r}(t) + h\vec{v}(t) + \frac{h^2}{2m} \vec{F}[\vec{r}(t)] + \mathcal{O}(h^3)$$

2 Velocities are determined from

$$\vec{v}(t+h) = \vec{v}(t) + \frac{h}{2m} [\vec{F}[\vec{r}(t+h)] + \vec{F}[\vec{r}(t)]] + \mathcal{O}(h^3)$$

- evaluates positions and velocities at the same time instances
- more stable than the original Verlet version with respect to errors from finite-precision arithmetic
- it may seem that the method needs two force arrays  $\vec{F}[\vec{r}(t)]$  and  $\vec{F}[\vec{r}(t+h)] \dots$

## Velocity Verlet algorithm: implementation

### 1 STEP 1:

$$\widetilde{\vec{v}(t)} = \vec{v}(t) + \frac{h}{2m} \vec{F} [\vec{r}(t)]$$

### 2 STEP 2:

$$\vec{r}(t+h) = \vec{r}(t) + h \widetilde{\vec{v}(t)}$$

### 3 STEP 3: calculate the new force $\vec{F} [\vec{r}(t+h)]$ and replace the array $\vec{F} [\vec{r}(t)]$ by $\vec{F} [\vec{r}(t+h)]$

### 4 STEP 4:

$$\vec{v}(t+h) = \widetilde{\vec{v}(t)} + \frac{h}{2m} \vec{F} [\vec{r}(t+h)]$$

### 5 STEP 5: go back to step 1 with $t \rightarrow t + h$

# Verlet algorithm in the program MD.F

All Verlet formulations have the same memory requirements! Program adopts the velocity-Verlet algorithm

$$\begin{aligned}x(t+h) &= x(t) + hv_x(t) + \frac{h^2}{2} F_x(t) + \mathcal{O}(h^3) \\v_x(t+h) &= v_x(t) + \frac{h}{2} [F_x(t+h) + F_x(t)] + \mathcal{O}(h^3)\end{aligned}$$

Positions: Qx(I), Qy(I), Qz(I), I=1, PartNum

Momenta: Px(I), Py(I), Pz(I), I=1, PartNum

**SUBROUTINE** Integrate (**ForceX, ForceY, ForceZ, Fac2, Scale, Virial**)

**SUBROUTINE** CalcForce(**ForceX, ForceY, ForceZ, Virial**)

## STEP 2: Start simulation and let system reach equilibrium (III)

- Lennard-Jones potential  $U(r) = \epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - 2 \left( \frac{\sigma}{r} \right)^6 \right]$  determines the forces between the particles
- Cutting off the force at  $r = r_{cut-off}$  violates energy conservation!
- Non-conservation of energy induced by discontinuities in the energy and potential can be remedied by shifting the potential (force and potential are made continuous)

$$\begin{aligned} U_{shift}(r) &= U(r) - U(r_{cut-off}) \\ &- \left[ \frac{d}{dr} U(r_{cut-off}) \right] (r - r_{cut-off}) \end{aligned}$$

### **SUBROUTINE** *CalcForce*

- Time needed to reach equilibrium depends on how far the initial configuration was from equilibrium and on the relaxation time of the system

## STEP 2: Start simulation and let system reach equilibrium (IV)

- To force the system to have a desired temperature  $T_D$

$$\vec{v}_i(t) \rightarrow \lambda \vec{v}_i(t) \quad \text{with,} \quad \lambda = \sqrt{\frac{(N-1)3kT_D}{\sum_{i=1}^N mv_i^2}}$$

- After each rescaling: system will drift away from  $T_D$ . After some time, however, temperature fluctuates about  $T_D$ !
- This simple trick of imposing a constant temperature  $T_D$  is only valid when computing physical variables which do not depend on momenta and/or time!

$$\langle A(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots, \vec{r}_N) \rangle$$

**SUBROUTINE** *CalcTemp(TotEner)*

**SUBROUTINE** *Rescale (TotEner)*

# How to determine the force on particle $i$ at time $t$ ? (I)

$\vec{F}[\vec{r}_i(t)] = \sum_{j \neq i} \vec{F}[\vec{r}_i(t) - \vec{r}_j(t)]$  is an expensive operation as it goes like  $\mathcal{O}(N^2)$ !

- $\sum_{j \neq i}$  is an infinite sum with PBC
- minimum image convention:
  - 1  $\sum_j$  gets reduced to a finite sum (nearest copy of each particle in the system)
  - 2 can be easily implemented

$$|x_1 - x_2| \rightarrow \left| |x_1 - x_2| - \left| \left[ \frac{|x_1 - x_2|}{\frac{L}{2}} \right] * L \right| \right|$$

$\left[ \right]$ : denotes the integer part

- 3 potential is no longer analytic: not problematic as long as  $U(r_{12} > \frac{L}{2}) \rightarrow 0$  ( $V = L^3$ )
- often one has  $U(r_{12} > r_{cut-off}) \rightarrow 0$  for  $r_{cut-off} < \frac{L}{2}$  : minimum image convention is no longer efficient

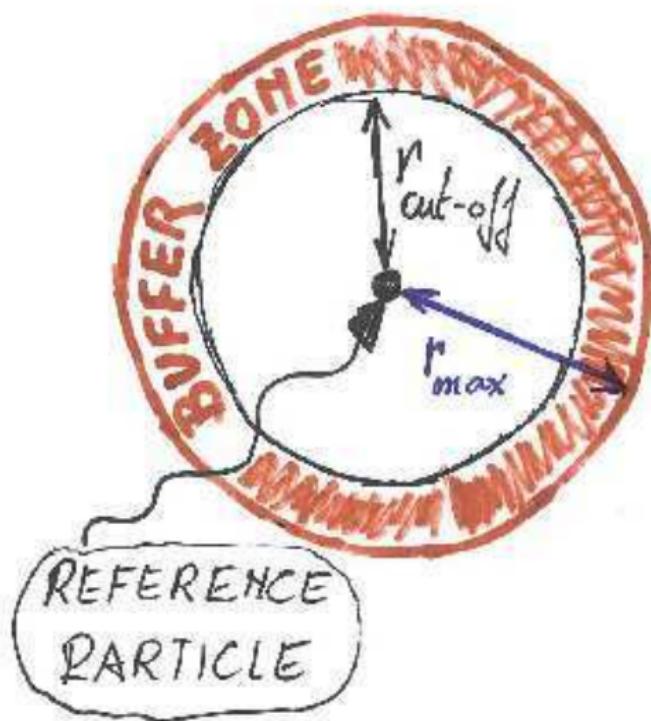
Methods to keep track of pairs within a certain distance of each other

- 1 Verlet method
- 2 linked-cell method

## Verlet method

- pair list for  $r_{12} \leq r_{max}$  with  $r_{max} > r_{cut-off}$
- update the list after typically 10 or 20 steps (pairs with  $r_{12} > r_{max}$  are unlikely to become closer than  $r_{cut-off}$ )
- method provides an efficiency gain of the order 10 and good accuracy

## How to determine the force on particle $i$ at time $t$ ? (III)



- \*  $r_{\text{cut-off}}$  : action radius of the force
- \*  $r_{\max} > r_{\text{cut-off}}$  : includes some buffer zone!

Methods to keep track of pairs within a certain distance of each other

- 1 Verlet method
- 2 linked-cell method

## linked-cell method

- system is divided up into rectangular cells of size  $r_{max} > r_{cut-off}$
- cell is characterized by  $(IX, IY, IZ)$
- particles leave and enter cells
- bookkeeping method for pairs in the same or neighbouring cells
  - 1 routine that links each particle to a cell
  - 2 force calculation must use this information

# How to determine the force on particle $i$ at time $t$ ? (V)

Method to link each particle to a cell ( $N$  is total number of particles,  $M \times M \times M$  cells in the system)

integer array Link (N) initialized to 0

integer array Header (M,M,M) initialized to 0

FOR I=1,N DO

IX = int(M\*x(I)/L) + 1

IY = int(M\*y(I)/L) + 1

IZ = int(M\*z(I)/L) + 1

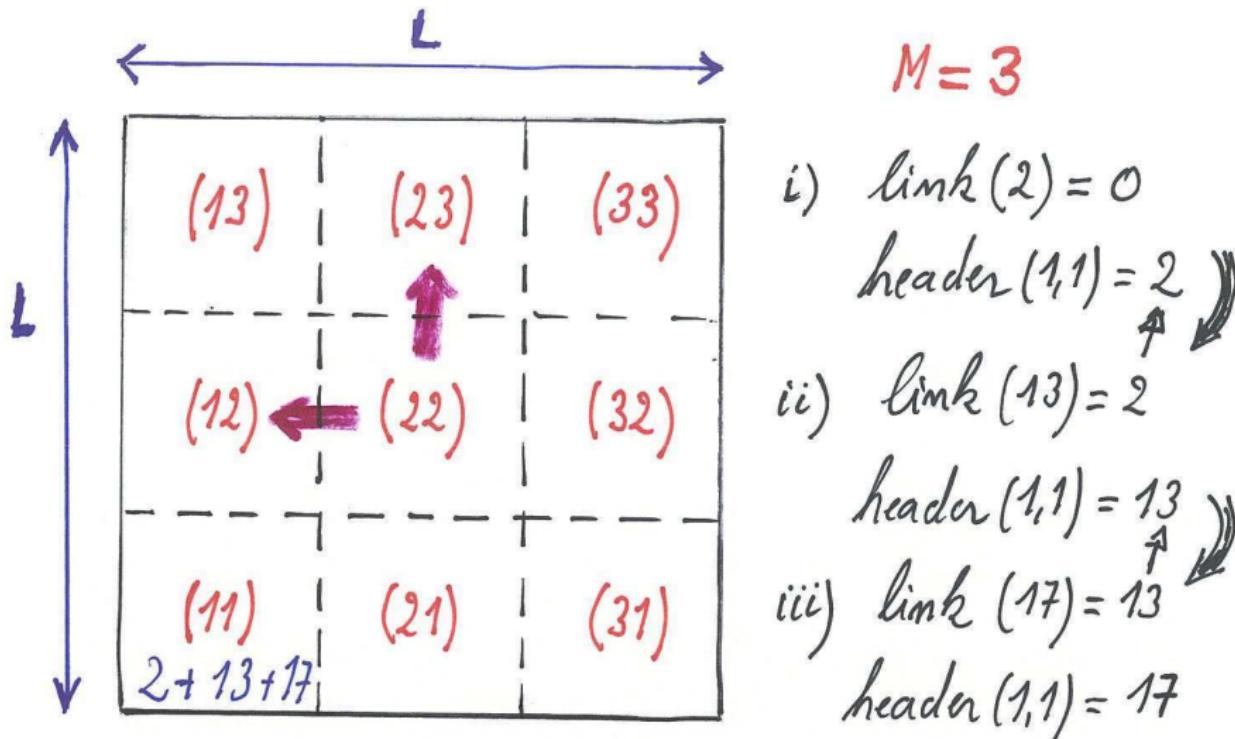
Link(I) = Header(IX,IY,IZ)

Header(IX,IY,IZ) = I

END FOR

- Header (IX,IY,IZ) contains highest index present in all cells
- Link (I) is another particle in the same cell

# How to determine the force on particle $i$ at time $t$ ? (VI)



**FORCE CALCULATION: ( $\Rightarrow$ )**

# How to determine the force on particle $i$ at time $t$ ? (VII)

Header(IX,IY,IZ) and Link(I) can be used for the force calculation

## Pseudocode for force calculation

FOR all cells with indices (IX,IY,IZ) DO

{Fill the list xt, yt and zt with the particles of the central cell}

  icnt = 0;

  j = Header(IX,IY,IZ);

  WHILE (j>0) DO

    j = link(j);

    icnt = icnt + 1;

    xt(icnt) = x(j); yt(icnt) = y(j); zt(icnt) = z(j);

    LocNum = icnt;

  END WHILE

{Now, LocNum is the number of particles in the central cell}

  FOR half of the neighbouring cells DO

    Find particles in the same way as central cell

    and append them to the list xt, yt, zt;

  END FOR

  Calculate Lennard-Jones forces between all particles in the central cell;

  Calculate Lennard-Jones forces between particles in central and

    neighbouring cells;

END FOR

- loop over half the number of neighbouring cells (to avoid double-counting)
- cell method is less efficient than the Verlet method (blocks versus spheres)
- advantage of the method lies in its suitability for parallel computing

# About units in MD simulations for Argon (I)

- Unit of mass: mass of Argon ( $^{40}_{18}\text{Ar}$ )  $m$   
( $m = 39.948 \text{ amu} = 39.948 \times 1.66 \times 10^{-27} \text{ kg}$ )
- Unit of energy:  $\epsilon \approx 0.01 \text{ eV}$  in Lennard-Jones potential. Fixes also the unit of temperature  $T = 1 \approx 120 \text{ K}$

$$\frac{\epsilon}{k_B} = 119.8 \text{ K} \quad k_B = 8.617343 \times 10^{-5} \text{ eV} \cdot \text{K}^{-1}$$

- Unit of distance :  $\sigma$  in Lennard-Jones potential  
( $\sigma = 3.822 \times 10^{-10} \text{ m}$ )
- Unit of time is fixed by units of mass, energy and distance

$$\tau = \sqrt{\frac{m\sigma^2}{\epsilon}} = 2.24 \times 10^{-12} \text{ s}.$$

## About units in MD simulations for Argon (II)

What is the proper time step  $\Delta t$  between two consecutive configurations?

- 1 Say that we simulate with 900 particles in two dimensions.
- 2 Accordingly, simulation system size  $L$  is of the order of  $L \approx 30 \times 10^{-10} \text{ m}$ .
- 3 Typical velocity of molecules in a gas  $v \approx 300 \text{ m s}^{-1}$ .
- 4 In one time-step  $\Delta t$  we let the molecules only move over  $\frac{1}{1000}$  of the system size:  $\Delta x \approx 30 \times 10^{-13} \text{ m}$
- 5 As a result, a properly chosen time step is of the order

$$\Delta t = \frac{\Delta x}{v} \approx 10^{-14} \text{ s}$$

In units adopted in the program:  $10^{-14} \text{ s} \approx 0.004 \tau$

# Appendix B: Random-number generators

## B.1 Random and pseudo-random numbers (I)

- Random numbers used in many simulations
  - e.g. understanding of particle accelerators: simulations whereby particles are injected in a random way
- Properties of random numbers
  - 1 value cannot be predicted
  - 2 independence: probability of a new number is independent of the numbers generated so far (NO CORRELATIONS!)
- Truly random numbers can be obtained through radioactive decay  
<http://www.fourmilab.ch/hotbits/>
- Random numbers generated on a computer are not truly random: new numbers are generated from previous ones  
**pseudo-random numbers** (not suitable for cryptographic purposes!)

## B.1 Random and pseudo-random numbers (II)

- Good pseudo-random number generators generate sequences which are statistically difficult to discern from sequences of pure random numbers
- Random numbers are uniformly distributed in  $[0, 1]$  ; distribution function  $P(x)$

$$\begin{aligned}P(x) &= 1 & 0 \leq x \leq 1 \\P(x) &= 0 & |x| > 1\end{aligned}$$

- $P(x_i, x_j)$  ( $j \geq (i + 1)$ ): probability for the successive occurrence of two random numbers in step  $i$  and step  $j$ .
- Absence of correlations is guaranteed for

$$P(x_i, x_j) = P(x_i)P(x_j) \quad (j \geq (i + 1))$$

## B.2 Random number generator (I)

- On a computer random numbers are represented by finite number of bits
- Thus, random numbers can be interpreted as integers
- $r$  bits leads to  $2^r$  possible numbers
- Mapping on  $[0, 1]$  by dividing by the largest integer that can be generated
- Two frequently used techniques:
  - 1 “linear congruent” or “modulo generator”
  - 2 shift-register generator

## B.2 Random number generator (II)

“linear congruent” or “modulo generator”

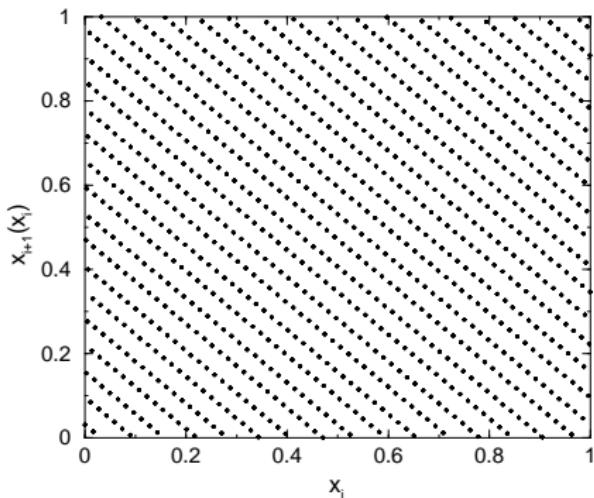
- modulo operation finds the remainder of the division by one number by another
- $$x_i = \left[ (ax_{i-1} + c) \bmod m \right] \quad (i > 0)$$
- Initial number  $x_0$  is the seed of the generator
- Mapping on  $[0, 1]$  by dividing by  $m - 1$  ( $m$  is a prime number)
- Very often:  $c = 0$ . The choice for the numbers  $a$  and  $m$  is a very subtle one! (e.g.  $a = 12$  and  $m = 143$  is a bad choice)
- To check the independence of the random numbers: take triples of random numbers and consider them as indices  $(x, y, z)$  of a point in three-dimensional space. These points should fill the unit cube homogeneously.

**Bad random generators abound! Check the results of any simulation with different random generators!**

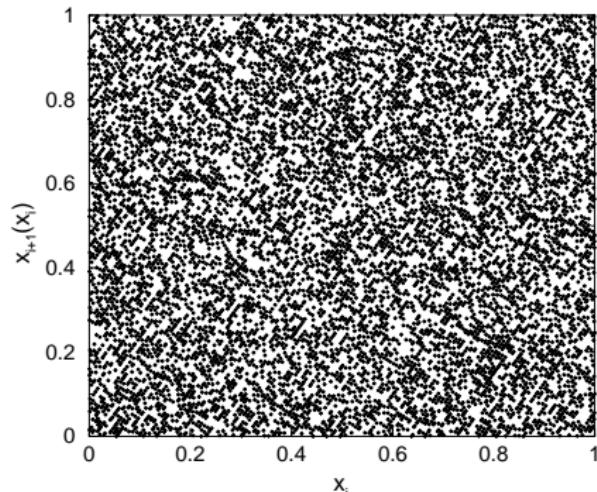
## B.2 Random number generator (III)

Correlations between successive random numbers  $x_i$  and  $x_{i+1}$  with modulo generator

**BAD ONE!!**



**GOOD ONE!!**



$$a = 12351; c = 1; m = 2^{15}$$

$$a = 12349; c = 1; m = 2^{15}$$

Source: PRACTICAL GUIDE TO COMPUTER SIMULATIONS by HARTMANN ALEXANDER K (World Scientific, 2009)

## B.2 Random number generator (IV)

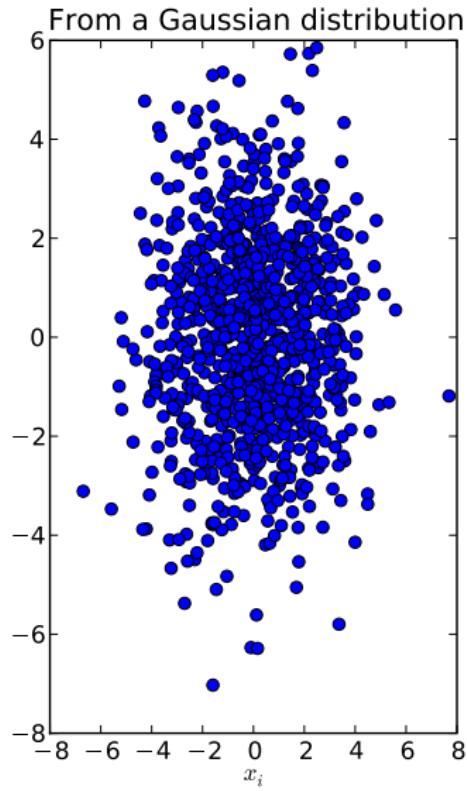
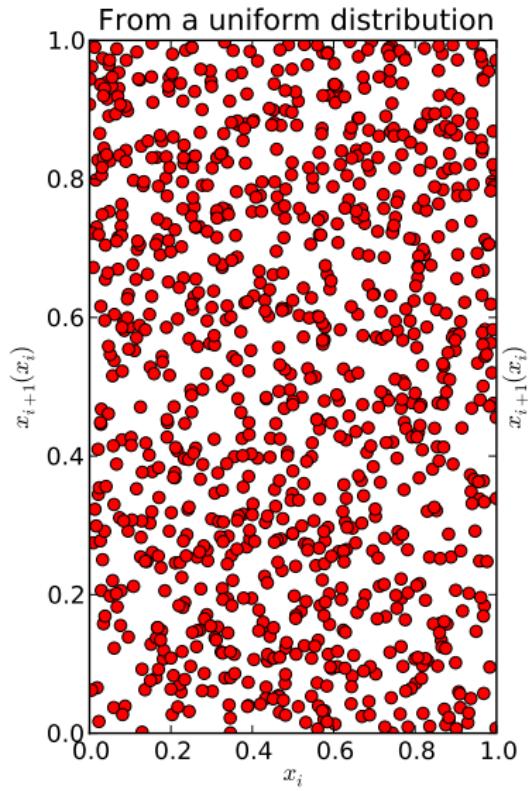
### Random number generators in Python

- `numpy.random` has a lot of functionalities!
- `random()`: a random float uniformly in the semi-open range  $[0.0, 1.0)$
- `random()`: 53-bit precision float with a period  $2^{19937} - 1$
- `random.normal( $\mu, \sigma$ , 1)`: draws one random number from a Gaussian with mean  $\mu$  and variance  $\sigma$
- Example: draw a series of random numbers from a uniform distribution and from a Gaussian nonuniform distribution and plot  $x_{i+1}$  versus  $x_i$

## B.2 Random number generator (V)

```
import pylab as pl
import numpy as np
#
#
howmany = int(input("How many random numbers would like to generate?"))
#
#
firstset = []
secondset = []
thirdset = []
fourthset = []
x=0
# random.random(): return the next floating point number in [0,1)
#
while x<=howmany:
    firstset.append(np.random.random())
    secondset.append(np.random.random())
#
# np.random.normal(0.,2.,1): draw one random number from a
#   Gaussian distribution with average 0. and variance 2.
#
    thirdset.append(np.random.normal(0.,2.,1))
    fourthset.append(np.random.normal(0.,2.,1))
    x=x+1
#
# start producing the plots
#
pl.subplot(121)
plot1 = pl.plot(firstset,secondset, 'ro')
pl.xlabel(r"$x_{i}$")
pl.ylabel(r"$x_{i+1} - (x_i)$")
pl.title('From a uniform distribution')
#
#
pl.subplot(122)
plot2 = pl.plot(thirdset,fourthset, 'bo')
pl.xlabel(r"$x_{i}$")
pl.ylabel(r"$x_{i+1} - (x_i)$")
pl.title('From a Gaussian distribution')
pl.show()
```

## B.2 Random number generator (VI)



## B.2 Random number generator (VII)

“shift-register generator”

- $k$ -th bit of the  $i$ -th number ( $\equiv b_i^{(k)}$ ) is determined from an expression of the type

$$b_i^{(k)} = (c_1 b_{i-1}^{(k)} + c_2 b_{i-2}^{(k)} + \dots + c_n b_{i-n}^{(k)} \bmod 2)$$

- The coefficients  $c_x$  are either 0 or 1!
- The above formulae requires the presence of  $n$  random numbers (modulo generator)
- Simple form of the generator if only two  $c_x$  are different from zero

$$b_i^{(k)} = b_{i-p}^{(k)} + b_{i-q}^{(k)} \bmod 2$$

- “Magical” ( $p, q$ ) pairs: (98,27), (521,32) and (250,103)
- The shift-register generator is superior to modulo generator!!!

## Random number generator with non-uniform distributions (Section B.3, p609) (I)

The momenta of the particles are initialised by drawing random numbers from a Maxwellian distribution! How can this be done?

- Number of uniformly distributed random numbers in  $[x, x + dx]$  is proportional to  $P(x)dx = dx$ , as  $P(x) = 1$  !
- Consider a variable  $y$  which is distributed according to  $P(y)$ . We ask ourselves, *Can we find a function  $f(x)$  such that  $x$  is uniformly distributed and  $y = f(x)$ ?*
- $P(y)dy$ : number of non-uniformly distributed random numbers in  $[y, y + dy]$ .
- Accordingly,  $P(y)$  is the density of numbers  $y$ . We have (remind that  $x = f^{-1}(y)$ )

$$P(y) = \frac{P(x)dx}{dy} = \frac{dx}{dy} = \frac{df^{-1}(y)}{dy} .$$

- The function  $f(x)$  exists if an invertible primitive function to the distribution  $P(y)$  can be found!!

## Random number generator with non-uniform distributions (Section B.3, p609) II

- It is not always possible to find an invertible primitive function to the distribution  $P(y)$ !
- A Gaussian distribution ( $G(y)$ ), for example, does not easily lend itself to this technique

$$G(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{y^2}{2\sigma^2} = \frac{d\mathbf{f}^{-1}(y)}{dy}$$

- The Maxwell distribution for the velocities in two dimensions is a good example of a distribution for which it is possible to find an invertible primitive function. Indeed,

$$y \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{y^2}{2\sigma^2} = \frac{d\mathbf{f}^{-1}(y)}{dy}$$

has a primitive function  $f(y)$  which is invertible.

# Random velocities from a Maxwell distribution (I)

- In equilibrium, a classical (ideal) gas in three dimensions has the following speed distribution

$$P(v)dv = \frac{4\pi m^3}{(\pi 2mk_B T)^{3/2}} v^2 \exp\left(-\frac{m}{2k_B T} v^2\right) dv$$

- As a result we can write for the vector  $\vec{v}$

$$\begin{aligned} P(\vec{v})d\vec{v} &= \frac{m^3}{(\pi 2mk_B T)^{3/2}} \exp\left(-\frac{m}{2k_B T} v^2\right) v^2 dv \sin \theta d\theta d\phi \\ &\equiv P(v_x)P(v_y)P(v_z)dv_x dv_y dv_z, \end{aligned}$$

where  $P(v_x)$ ,  $P(v_y)$  and  $P(v_z)$  are Gaussians

$$\left( G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{x^2}{2\sigma^2} \right)$$

$$P(v_x) = \frac{m}{\sqrt{2\pi m k_B T}} \exp\left(-\frac{m}{2k_B T} v_x^2\right)$$

## Random velocities from a Maxwell distribution (II)

Two-dimensional case:  $dv_x dv_y \equiv v dv d\phi$  and  $v^2 = v_x^2 + v_y^2$  gives rise to

$$\begin{aligned} P(v_x, v_y) dv_x dv_y &= \frac{m^2}{2\pi m k_B T} v \exp\left(-\frac{m}{2k_B T} (v_x^2 + v_y^2)\right) dv d\phi \\ &\equiv P_2(v) dv d\phi = P(v_x) P(v_y) dv_x dv_y \end{aligned}$$

With,

$$P_2(v) dv \sim v \exp\left(-\frac{v^2}{2}\right) dv = \frac{df^{-1}(v)}{dv} dv$$

Therefore,  $f^{-1}(v) = Cte - \exp\left(\frac{-v^2}{2}\right) \equiv x$  and  $v = \sqrt{-2 \ln(Cte - x)}$  with  $x$  a uniformly distributed random number in  $[0, 1]$ . With  $Cte = 1$  and the replacement  $1 - x \rightarrow x$  (preserving the interval  $[0, 1]$  of allowed values of  $x$ ) one gets

$$v = \sqrt{(-2) \ln x} \quad \text{with } x \text{ randomly distributed in } [0, 1] .$$

# Random velocities from a Maxwell distribution (III)

The initialisation of the momenta by randomly drawing from a Maxwell distribution is done with the aid of the following routines

***SUBROUTINE InitMomenta()***

***INTEGER FUNCTION RandGen()***

***DOUBLE PRECISION FUNCTION RealRand()***

***SUBROUTINE Ran(JRand)***

***SUBROUTINE InitRand(InitJ)***

***SUBROUTINE ExpRand(R1, R2)***

- 1 *RealRand()*: determines a uniformly distributed Random Number in  $[0, 1]$
- 2 *ExpRand(R1, R2)*: the  $R1$  and  $R2$  are random and distributed according to

$$\sqrt{(R_1^2 + R_2^2)} \exp \frac{-(R_1^2 + R_2^2)}{2}$$

# Random velocities from a Maxwell distribution (IV)

The subroutine *ExpRand(R1,R2)* adopts the Box-Müller method

The Box-Müller method: generate two numbers  $R_1$  and  $R_2$  which are both distributed according to a Gaussian

- 1 create random number  $x_1$  in  $[0, 1]$
- 2 define  $\text{Phi} = x_1 \times 2\pi$
- 3 create random number  $x_2$  in  $[0, 1]$
- 4 define  $\text{Norm} = \sqrt{-2 \ln x_2}$
- 5

$$R_1 = \text{Norm} \times \cos \text{Phi} \quad (v_x = v \cos \phi)$$

$$R_2 = \text{Norm} \times \sin \text{Phi} \quad (v_y = v \sin \phi)$$

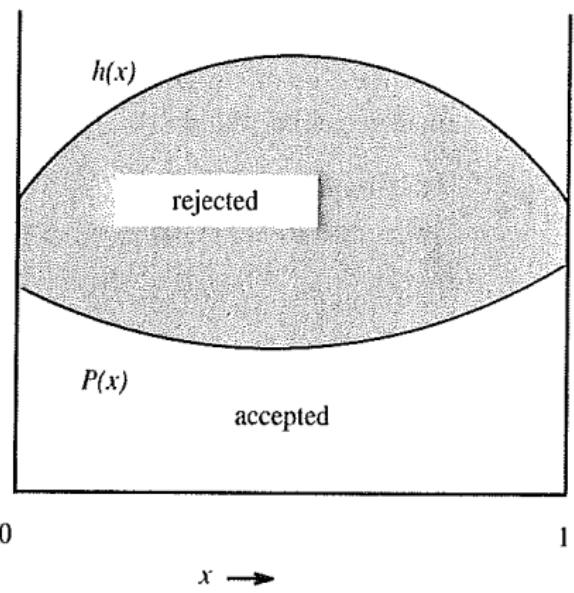
# Random velocities from a Maxwell distribution (IV)

The subroutine *InitMomenta* adopts the following strategy  
(Momenta of the particles  $P_x(I)$ ,  $P_y(I)$ ,  $P_z(I)$ ,  $I=1$ ,  
PartNum )

- 1 Loop over the particles 1,3,5, ...
- 2 Call to *ExpRand(R1,R2)*
- 3  $P_x(I)=R1$  and  $P_y(I)=R2$
- 4 Call to *ExpRand(R1,R2)*
- 5  $P_z(I)=R1$  and  $P_x(I+1)=R2$
- 6 Call to *ExpRand(R1,R2)*
- 7  $P_y(I+1)=R1$  and  $P_z(I+1)=R2$

# If one cannot find a primitive function ...

Method of Von Neumann for generating nonuniform random numbers  $P(x)$  in  $[a, b]$



- at least two uniform random numbers to generate a single nonuniform one
- works for  $P(x)$  that fits in a box
- choose a distribution  $h(x)$  with  $h(x) > \alpha P(x)$  in  $[a, b]$
- $h$  can be the uniform generator ; more efficient to have  $\frac{\alpha P(x)}{h(x)} \approx 1$
- for every  $x$  generated with  $h(x)$  one generates a  $y$  with a uniform generator in  $[0, 1]$
- for  $y < \frac{\alpha P(x)}{h(x)}$  one accepts  $x$

## STEP 3: Determine the physical quantities (I)

- General definition for the correlation function: joint probability to find a particle at  $\vec{r}$  AND a particle at  $\vec{r}'$

$$g(\vec{r}, \vec{r}') = \frac{V^2}{N(N-1)} \left\langle \sum_{i \neq j=1}^N \delta(\vec{r} - \vec{r}_i) \delta(\vec{r}' - \vec{r}_j) \right\rangle$$

- In a uniform system: correlation function for particles that are  $\vec{r}_{12}$  apart ( $g(\vec{r}, \vec{r}')$  depends only on the relative distance between the two tagged particles)

$$g(\vec{r}_{12}) = \frac{V}{N(N-1)} \left\langle \int d\vec{r}' \sum_{i \neq j=1}^N \delta(\vec{r}' - \vec{r}_i) \delta(\vec{r}' + \vec{r}_{12} - \vec{r}_j) \right\rangle$$

## STEP 3: Determine the physical quantities (II)

- In MD simulation for Argon determine the histogram  $n(r_{12})$  : average number of pairs with separation into the range  $[r_{12}, r_{12} + \Delta r_{12}]$ .
- Correlation function  $g(r_{12})$  is related to  $n(r_{12})$

$$g(r_{12}) = \frac{2V}{N(N-1)} \left[ \frac{n(r_{12})}{4\pi r_{12}^2 \Delta r_{12}} \right]$$

- Due to the periodic boundary conditions:  $g(r_{12})$  can be determined for  $r_{12} \leq \frac{L}{2}$  ( $L$  is the linear system size)

## STEP 3: Determine the physical quantities (III)

With the pair correlation function  $g(r_{12})$  accurate values for potential energy and pressure can be obtained!

- Pressure equation

$$\frac{P}{\rho k_B T} = 1 - \frac{\rho}{6k_B T} \int_0^{+\infty} dr_{12} \frac{dU(r_{12})}{dr_{12}} g(r_{12}, \rho, T) 4\pi r_{12}^3$$

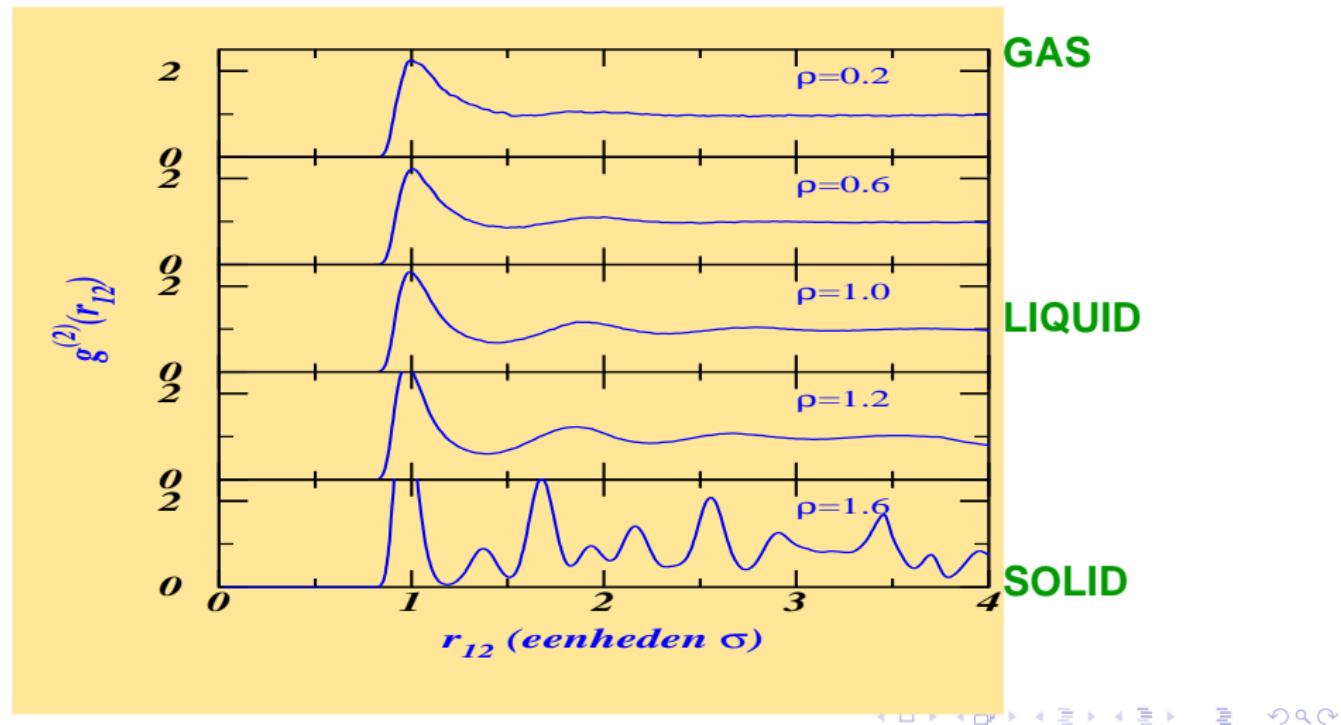
- The neglect of the tail of the potential can be corrected for

$$\langle U \rangle = \langle U \rangle_{cut-off} + 2\pi \frac{N(N-1)}{V} \int_{r_{cut-off}}^{\infty} dr r^2 U(r) g(r)$$

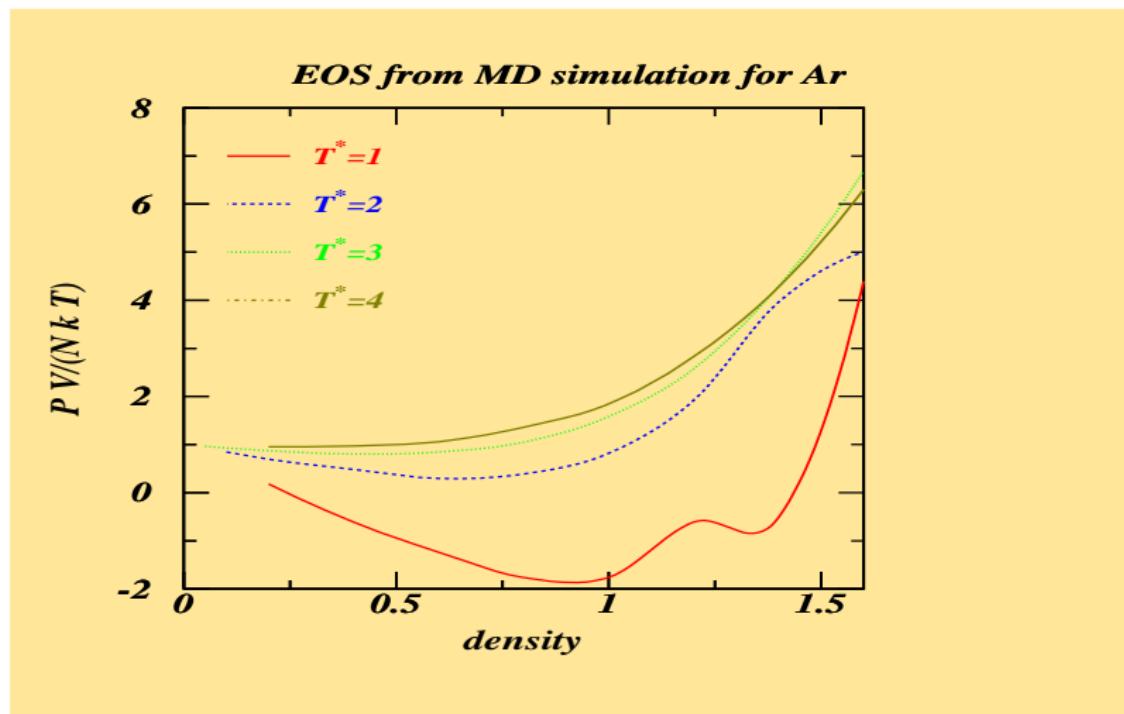
$\langle U \rangle_{cut-off}$  is the average restricted to pairs with separation  $r_{12} < r_{cut-off}$

# Density dependence of correlation function

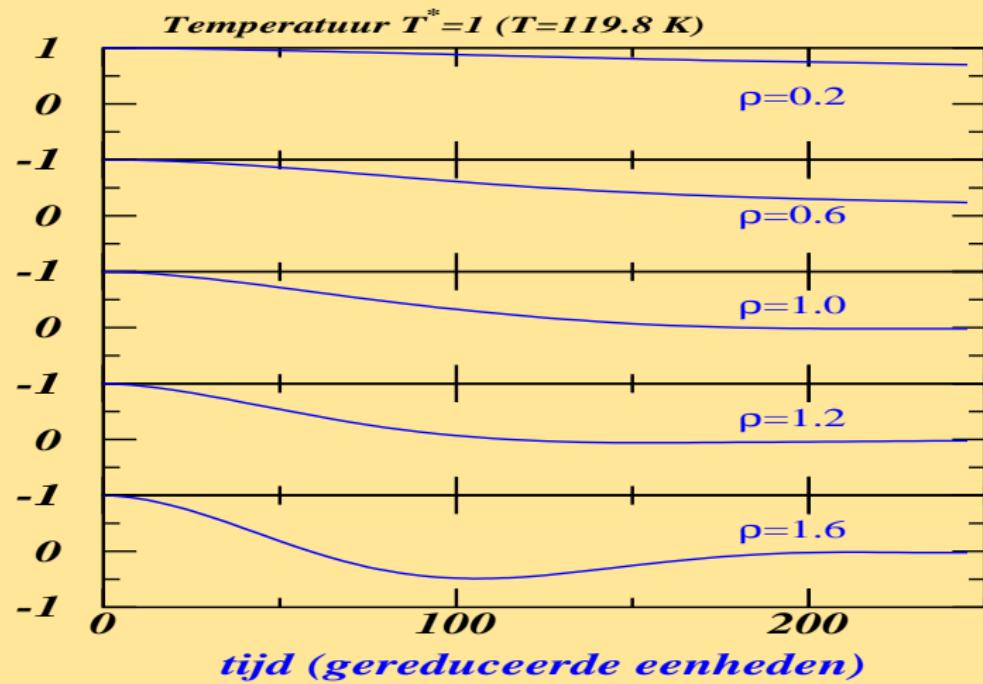
Simulation with 500 molecules at  $T^* = 1 = 119.8$  K



# Equation of state: the pressure as a function of the density and the temperature



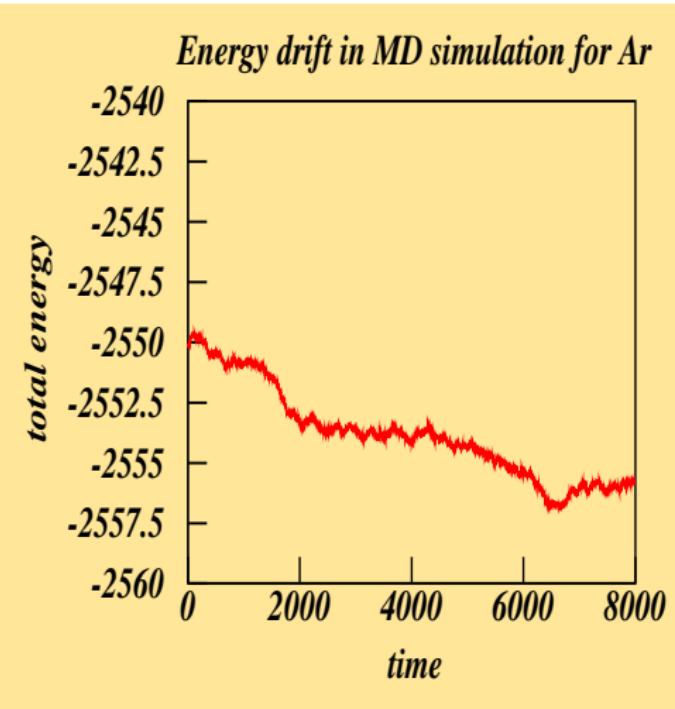
# Density dependence of velocity autocorrelation function



## 8.4 Integration methods - symplectic integrators

- Many algorithms for integrating ordinary differential equations (Appendix A)
- Verlet algorithm is very popular for MD simulations
- A few criteria to consider when selecting an algorithm
  - 1 The accuracy is the criterion in numerical analysis (e.g.  $\mathcal{O}(h^3)$ ), but ... prefactor of  $\mathcal{O}$  may diverge.
  - 2 Along the physical trajectory the energy is conserved! The numerical trajectory will have a drift (steady increase or decrease) and noise fluctuations on top of the drift. A system which drifts away from a constant-energy surface is NOT in equilibrium.
- Verlet algorithm is not susceptible to energy-drift in exact arithmetic!
- In practice: energy drift as a result of finite precision of computer arithmetic.
- Energy-drift can be decreased by making  $\Delta t$  smaller ... requires more computing time.

# Energy drift in an MD simulation for Argon



- Drift in energy due to finite precision of computer arithmetic!
- Source 1: truncation of numbers “cools” the system ; rounding off of numbers acts as small random interactions (“heats” the system)
- Source 2: algorithm to solve differential equation

TimeStep=0.0005  
(or,  $\Delta t = 1.12 \times 10^{-15}$  s).

## 8.4.1 The Verlet algorithm revisited (I)

### Verlet algorithm to determine positions

- Start from the following Taylor expansion

$$x(t \pm h) = x(t) \pm hv_x(t) + \frac{h^2}{2} F_x \pm \frac{h^3}{3!} \ddot{x}|_t + \mathcal{O}(h^4)$$

(units are chosen such that  $m = 1$ ).

- Combination of the two Taylor expansions leads to

$$x(t+h) = 2x(t) - x(t-h) + h^2 F_x + \mathcal{O}(h^4)$$

- After a number of integrations, error on the positions is of the order  $\mathcal{O}(h^2)$ .

## 8.4.1 The Verlet algorithm revisited (II)

### Verlet algorithm to determine velocities

- In principle it suffices to determine the velocity through

$$v_x(t) = \frac{x(t+h) - x(t-h)}{2h} + \mathcal{O}(h^2) \quad (E2)$$

- “Improved” version (E3):

$$v_x(t) = \frac{x(t+h) - x(t-h)}{2h} - \frac{h}{12} \left[ F_x(t+h) - F_x(t-h) \right] + \mathcal{O}(h^3)$$

- LEAP-FROG VERSION (velocities at times halfway between those at which positions are determined)

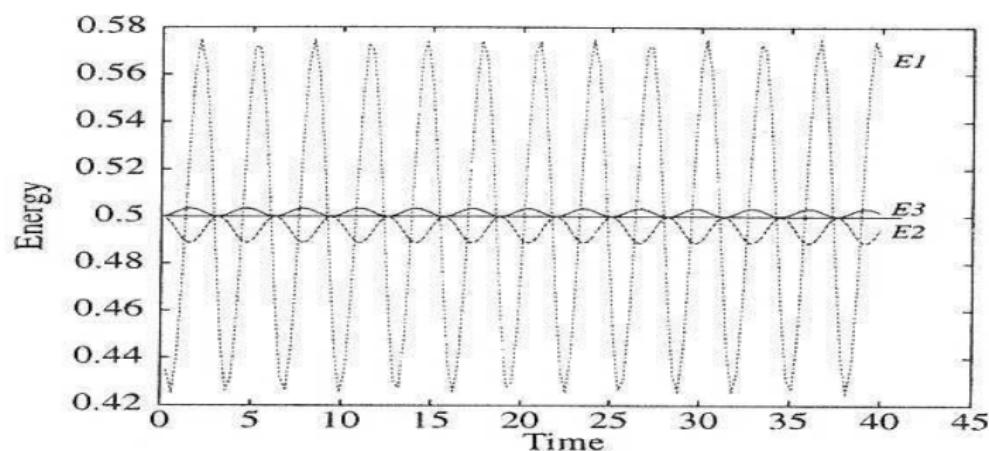
$$v_x \left( t + \frac{h}{2} \right) = \frac{[x(t+h) - x(t)]}{h} + \mathcal{O}(h^2) \quad (E1)$$

## 8.4.1 The Verlet algorithm revisited (III)

For the Harmonic oscillator (period  $P = 2\pi$ ):

$$E(t) = \frac{(v_x(t))^2 + (x(t))^2}{2}$$

Determine the energy with the aid of the energy estimators  $E1$ ,  $E2$  and  $E3$  for an integration time step of 0.3



## 8.4.1 The Verlet algorithm revisited (IV)

- $x(t)$  and  $x(t - h)$  are needed to determine  $x(t + h)$  (method is not self starting). Knowledge of  $x(0)$  and  $v(0)$  allows one to determine  $x(h)$  through (Euler method)

$$x(h) = x(0) + hv_x(0) + \frac{h^2}{2} F_x(0) + \mathcal{O}(h^3)$$

- Remark that the Verlet algorithm is time-reversible (invariant with respect to  $h \leftrightarrow -h$  and  $v_x \leftrightarrow -v_x$ ). After all, time-reversal is at the origin of energy conservation!
- Verlet algorithm possesses the “symplecticity” symmetry.
  - 1 Symplecticity gives rise to conserved quantities: for example a discrete analogue of the total energy is rigorously conserved (in exact arithmetic).
  - 2 Deviation between the discrete energy and the continuum energy is of the order  $h^k$ ,  $k \in \mathcal{N}$  (under control!).

## 8.4.1 The Verlet algorithm revisited (V)

- It is tempting to use  $\mathcal{O}(h^k)$  to classify one method as superior or inferior to another. Such classification is inappropriate as each method has its strengths and weaknesses. Each problem has a different optimum method!
- In recent years, major breakthroughs in understanding the merits of the various methods for integrating equations-of-motion (e.g. the concept of symplectic geometry).
- The strength of the Verlet algorithm in conserving the energy (micro-canonical ensemble!) can be well tested for a harmonic oscillator. For  $k = m = 1$  the period is  $2\pi$  and the total energy

$$E = \frac{1}{2} (x^2 + v^2)$$

# Verlet algorithm and harmonic oscillator (Fortran)

```
program verletho
ccc one-dimensional HO with Original Verlet algorithm
ccc
dimension x(0:20000),v(0:20000)
write(*,*) '_value_for_initial_position'
read(*,*) x(0)
write(*,*) '_value_for_timestep'
read(*,*) deltat
write(*,*) '_value_for_initial_velocity'
read(*,*) v(0)
write(*,*) '_value_of_the_string_constant'
read(*,*) xk
ccc
ccc determine x(1) with the Euler algorithm
ccc
d2=deltat**2/2.
h2=deltat**2
x(1)=x(0)+deltat*v(0)-d2*xk*x(0)
total_energy=0.5*(v(0)**2+xk*x(0)**2)
      write(4,102) 0.,x(0),v(0),total_energy
102      FORMAT(1x,F8.3,1x,F8.3,1x,F8.3,1x,F8.7)
ccc
ccc Position and velocity with original Verlet algorithm
ccc
do jj=2,20000
  x(jj)=2*x(jj-1)-x(jj-2)-h2*xk*x(jj-1)
  v(jj-1)=(x(jj)-x(jj-2))/(2.*deltat)
  total_energy = 0.5 * (v(jj-1)**2+xk*x(jj-1)**2)
  write(4,102) (jj-1)*deltat,x(jj-1),v(jj-1),total_energy
enddo
stop
end
```

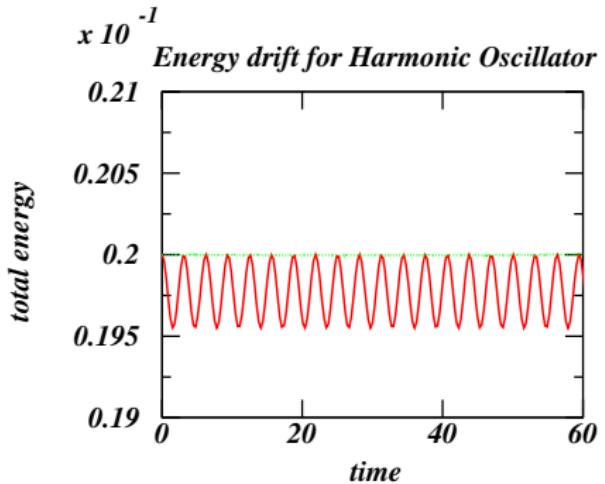
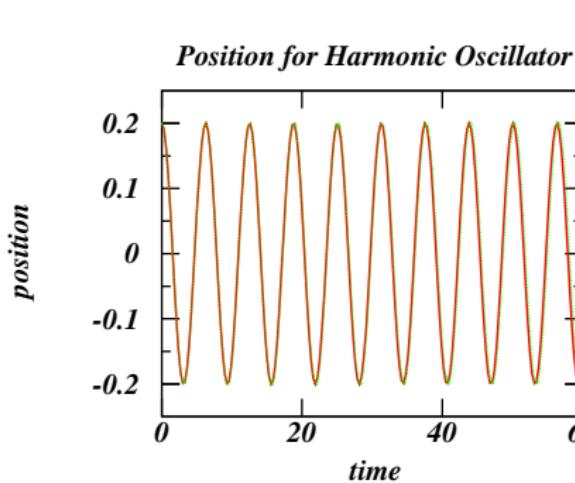
# Verlet algorithm and harmonic oscillator (Python)

```
import pylab as pl
import numpy as np
# position, velocity, time, total energy
x = []
v = []
tt = []
total_energy = []
# position and velocity at t=0
x.append(input("Position at t=0:"))
v.append(input("Velocity at t=0:"))
# time step and string constant
deltat=input("Provide the time step h:")
xk = input("Provide the string constant:")
# total energy and time at t=0
total_energy.append(0.5*(pow(v[0],2)+xk+pow(x[0],2)))
tt.append(0.+0.)
#
h2=pow(deltat,2)
d2=h2/2.
#
# determine x[t=1] with the Euler algorithm
#
x.append(x[0]+deltat*v[0]-d2*xk*x[0])
tt.append(1.*deltat)
#
# Position and velocity with original Verlet algorithm
#
jj = 2
while (jj+deltat)<=60.:
    x.append(2*x[jj-1]-x[jj-2]-h2*xk*x[jj-1])
    tt.append(jj+deltat)
    v.append((x[jj]-x[jj-2])/(2.*deltat))
    total_energy.append(0.5 + (pow(v[jj-1],2)+xk+pow(x[jj-1],2)))
    jj = jj + 1
# velocity and total_energy is one index behind ...
v.append((x[jj-1]-x[jj-3])/(2.*deltat))
total_energy.append(0.5 + (pow(v[jj-2],2)+xk+pow(x[jj-2],2)))
# THE PLOTS ...
plot1 = pl.plot(tt,x,'r-')
pl.xlabel(r"time", fontsize=20)
pl.ylabel(r"position", fontsize=20)
pl.ylim(1.10-min(x),1.10+max(x))
pl.title('Harmonic Oscillator with original Verlet: $\Delta t = $'+str(deltat), fontsize=18)
pl.show()
#
plot2 = pl.plot(tt, total_energy, 'r-')
pl.xlabel(r"time")
pl.ylabel(r"total energy")
pl.ylim(0.95-min(total_energy),1.05+max(total_energy))
pl.title('Harmonic Oscillator with original Verlet: $\Delta t = $'+str(deltat), fontsize=18)
pl.show()
```

# Results for Harmonic Oscillator

Result from VERLETHO for  $v(0) = 0.0$ ,  $xk = 1.$ ,  $x(0) = 0.2$

Green dashed line:  $deltat = 0.01$  ; red solid line:  $deltat = 0.3$ .



Energy fluctuation follows the periodicity of the motion! Energy fluctuation is almost non-existent for small time steps!

## 8.4.1 The Verlet algorithm revisited (VI)

### Inventory of relevant symmetry properties of integrators

- time reversibility!
- phase-space conservation (Liouville's theorem):

$$\begin{aligned}\rho(\vec{R}, \vec{P}) &= \rho(q_1, q_2, \dots, q_{3N}, p_1, p_2, \dots, p_{3N}) \\ \frac{d\rho}{dt} &= \frac{\partial\rho}{\partial t} + \sum_{i=1}^{i=3N} \left( \frac{\partial\rho}{\partial q_i} \frac{dq_i}{dt} + \frac{\partial\rho}{\partial p_i} \frac{dp_i}{dt} \right) = 0\end{aligned}$$

- symmetry requirement: symplecticity (combination of phase-space conservation and a number of conserved quantities, that are called Poincaré invariants)
- in practice: total energy should fluctuate around the exact one!
- safest option in order to conserve energy: use of symplectic integrations

## 8.4.1 Verlet algorithm with frictional forces (p216-) I

- Consider equation-of-motion of the type:  $\ddot{x} = F(x) - \gamma \dot{x}$  .
- Taylor expansion for  $x(\pm h)$  and  $\dot{x}(0)$

$$x(\pm h) = x(0) \pm h \dot{x}(0) + \frac{h^2}{2} [-\gamma \dot{x}(0) + F(0)] \pm \frac{h^3}{6} \ddot{x}(0) + \mathcal{O}(h^4)$$
$$\dot{x}(0) = \frac{x(h) - x(-h)}{2h} + \mathcal{O}(h^2)$$

- Through combination of the above expressions:

$$\left(1 + \frac{\gamma h}{2}\right) x(h) = 2x(0) - \left(1 - \frac{\gamma h}{2}\right) x(-h) + h^2 F(0) + \mathcal{O}(h^4) .$$

Or, knowledge of  $x(0)$  and  $x(-h)$  allows one to determine  $x(h)$ .

## 8.4.1 Verlet algorithm with frictional forces (p216-) II

- Leap-frog versions of the Verlet algorithm are based on the following principles
  - 1 The positions are known at time step  $h, 2h, 3h, 4h, \dots$
  - 2 The velocities are known at the time steps  $\frac{h}{2}, \frac{3h}{2}, \frac{5h}{2}, \frac{7h}{2}, \dots$
- Leap-frog version of the Verlet algorithm when including frictional forces
  - 1 VELOCITIES:  $p\left(\frac{h}{2}\right) = \frac{x(h) - x(0)}{h} + \mathcal{O}(h^2)$
  - 2 POSITIONS:

$$\begin{aligned}\ddot{x}(0) &= \frac{p\left(\frac{h}{2}\right) - p\left(-\frac{h}{2}\right)}{h} = F(0) - \gamma \dot{x}(0) \\ &= F(0) - \gamma \left[ \frac{p\left(\frac{h}{2}\right) - p\left(-\frac{h}{2}\right)}{2} \right]\end{aligned}$$

## 8.4.1 Verlet algorithm with frictional forces (p216-) III

- The above can be rewritten in a leap-frog version of the Verlet algorithm when the system is subject to forces  $F(x)$  AND a friction  $-\gamma\dot{x}$

- 1 Momenta (velocities) can be determined from

$$p\left(\frac{h}{2}\right) = \frac{\left(1 - \frac{\gamma h}{2}\right) p\left(-\frac{h}{2}\right) + hF(0)}{1 + \frac{\gamma h}{2}}$$

**knowledge of  $p\left(-\frac{h}{2}\right)$  and  $F(0)$  determines  $p\left(\frac{h}{2}\right)$**

- 2 Positions can be determined from

$$x(h) = x(0) + hp\left(\frac{h}{2}\right)$$

**knowledge of  $p\left(\frac{h}{2}\right)$  and  $x(0)$  determines  $x(h)$**

## 8.5 Molecular dynamics methods for different ensembles

- Experimental situations:  $T$  is a control variable, for  $E$  this is often not the case
- In an infinite system

$$\langle E_{kin} \rangle = \frac{3}{2}(N - 1)k_B T .$$

Method is used to compute  $T$  in a finite system.

- In MD simulations: energy  $E$  is a constant

**How we can perform MD simulations at constant temperature  $T$  or pressure  $P$ ?**

## 8.5.1 Constant temperature

Experience from real life is a useful guide to understand procedures for keeping temperature at a constant value

- In real life: system exchanges heat with a heat bath  
*microscopically through collisions between particles in the heat bath with particles at the surface in the system under study*
- In simulations: allow heat to flow to and from the system. Ideally, this leads to a canonical distribution of the configurations

$$\rho(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N, \vec{p}_1, \vec{p}_2, \dots, \vec{p}_N) \equiv \rho(\vec{R}, \vec{P}) = \exp - \frac{\mathcal{H}(\vec{R}, \vec{P})}{k_B T}.$$

*in simulations any particle can couple to the heat bath*

# Overview of the canonical MD methods

## HOW TO IMPOSE A DESIRED TEMPERATURE $T$ ???

### 1 Canonical MD method 1: Andersen method

- replace particle velocities every now and then
- optimum rate? loss of time correlations?

### 2 Canonical MD method 2: rescaling method

- rescaling of the velocities such that  $\langle E_{kin} \rangle = \frac{3(N-1)k_B T}{2}$
- only canonical in the coordinates!

### 3 Canonical MD method 3: frictional force

- introduce a friction term in the equations-of-motion
- how to determine the friction term?

# Canonical MD method 1: Andersen method

In 1980 Andersen devised a method in which the temperatures is kept constant by replacing every so often the velocity of a randomly chosen particle by a velocity drawn from a Maxwell distribution with the desired temperature.

- Close to the experimental situation!
- Desired temperature can be reached quickly through a high rate ... but destroys long time tail of velocity autocorrelation function
- If rate is low, the equilibrium will be very slow
- Optimum rate for given density  $n$  and number of particles  $N$  is given by (problem 8.6)

$$R_{\text{collisions}} \sim \frac{\kappa}{k_B n^{1/3} N^{2/3}} \quad (\text{with } \kappa : \text{THERMAL CONDUCTIVITY})$$

- Andersen's method leads to a canonical distribution for all  $N$  and finds applications in lattice field theories

## Canonical MD method 2: rescaling method

- When determining expectation values of variables which are only function of  $\vec{R}$  only the canonical distribution in the positional coordinates is required

$$\rho(\vec{R}) = \exp - \frac{U(\vec{R})}{k_B T} .$$

- This can be enforced by replacing the distribution function for the kinetic energy

$$\rho(E_{kin}) \rightarrow \delta \left( E_{kin} - \frac{(3N - 3) k_B T}{2} \right) .$$

- This velocity rescaling procedure induces deviations from the canonical distribution of order  $1/\sqrt{N}$ .

**This rescaling method is rather ad hoc! (but it is simple one)**

## Canonical MD method 3: frictional force (I)

- keeping the temperature constant requires a coupling to a heat bath
- this coupling can be imposed by adding a frictional force to the system with the purpose of keeping the temperature (or,  $\langle E_{kin} \rangle$ ) constant

$$m\ddot{\vec{r}}_i = \vec{F}_i(\vec{R}) - \zeta(\vec{R}, \dot{\vec{R}})\dot{\vec{r}}_i \quad \vec{R}(\vec{r}_i, i = 1, \dots, N) .$$

the friction term is independent of  $i$  !

- $\zeta > 0$ : heat must be drained from the system
- $\zeta < 0$ : heat must be added to the system
- various forms for the frictional force  $\zeta(\vec{R}, \dot{\vec{R}})$  have been proposed

## Canonical MD method 3: frictional force (II)

### Various forms for $\zeta(\vec{R}, \dot{\vec{R}})$

- 1 The function  $\zeta(\vec{R}, \dot{\vec{R}})$  is determined by

$$\frac{d\langle E_{kin} \rangle}{dt} = 0 \quad \text{or,} \quad \zeta(\vec{R}, \dot{\vec{R}}) = -\frac{\frac{d\langle V(\vec{R}) \rangle}{dt}}{\sum_i \dot{\vec{r}}_i^2}$$

Resulting distribution is purely canonical in the coordinate part only!

- 2 Method proposed by Berendsen

$$\zeta = \gamma \left( 1 - \frac{T}{T_D} \right) \quad \text{with,} \quad T = \frac{\sum_{i=1}^{i=N} m v_i^2}{3Nk_B}$$

Temperature decays to the desired temperature exponentially with time at a rate determined by  $\gamma$ .

- 3 Nosé-Hoover thermostat

# About canonical MD methods

- **IN THEORY**

- Andersen and Nosé-Hoover method lead to a canonical distribution

- **IN REAL LIFE**

- very often the temperature levels down at the wrong temperature
- very often one observes fluctuations in  $T$  that are substantially larger than expected

$$\frac{\overline{\Delta T}}{T} = \sqrt{\frac{2}{ND}}$$

- $D$  is the dimension of the system
- relation between  $T$  and  $\overline{T_{kin}}$

$$\overline{T_{kin}} = \frac{DN}{2} k_B T$$

- $\overline{\Delta T}$  can be computed from

$$\overline{(\Delta T_{kin})^2} = \overline{T_{kin}^2} - \overline{T_{kin}}^2$$

- things should be checked carefully!

## 8.5.1 The Nosé-Hoover thermostat (p 227) (I)

- Simulation of a system defined by the Hamiltonian  $\mathcal{H}_0(\vec{R}', \vec{P}')$

$$\mathcal{H}_0(\vec{P}', \vec{R}') = \sum_{i=1}^{i=N} \frac{\vec{p}'_i{}^2}{2m} + \sum_{i < j} U(\vec{r}'_i - \vec{r}'_j)$$

- IDEA: Perform the simulation with a well-chosen Hamiltonian with  $6N + 2$  degrees of freedom  
**(or, include the heat bath explicitly)**

$$\begin{aligned} \mathcal{H}(\vec{P}, \vec{R}, p_s, s) &= \sum_{i=1}^{i=N} \frac{\vec{p}_i{}^2}{2ms^2} + \sum_{i < j} U(\vec{r}_i - \vec{r}_j) \\ &+ \frac{p_s^2}{2Q} + (3N + 1)k_B T \ln s. \end{aligned}$$

- The quantities  $(\vec{R}, \vec{P})$  are virtual variables which can be related to the “real” coordinates  $\vec{R}' (\vec{r}'_i, i = 1, \dots, N)$  and  $\vec{P}' (\vec{p}'_i, i = 1, \dots, N)$

## 8.5.1 The Nosé-Hoover thermostat (p 227) (II)

- Well-chosen and “fictitious” dynamical variables  $(s, p_s)$  are introduced.
- The  $(s, p_s)$  rescale time and kinetic energy in order to reach the conditions of a canonical distribution in the simulation
- Equations of motion for the Hamiltonian  $\mathcal{H}(\vec{P}, \vec{R}, p_s, s)$  in terms of the virtual variables

$$\left\{ \begin{array}{lcl} \frac{d\vec{r}_i}{dt} & = & \frac{\partial \mathcal{H}}{\partial \vec{p}_i} = \frac{\vec{p}_i}{ms^2} \\ \frac{ds}{dt} & = & \frac{\partial \mathcal{H}}{\partial p_s} = \frac{\vec{p}_s}{Q} \\ \frac{d\vec{p}_i}{dt} & = & -\frac{\partial \mathcal{H}}{\partial \vec{r}_i} = -\sum_{i < j} \vec{\nabla}_i U(\vec{r}_i - \vec{r}_j) \\ \frac{dp_s}{dt} & = & -\frac{\partial \mathcal{H}}{\partial s} = \frac{\sum_i \vec{p}_i \cdot \vec{p}_i}{2ms^2} - \frac{(3N+1)k_B T}{s} \end{array} \right.$$

For  $N$  particles in three dimensions:  $(6N + 2)$  ODEs

## 8.5.1 The Nosé-Hoover thermostat (III)

- With the aid of

$$\delta(f(x)) = \frac{\delta(x - x_0)}{\left. \frac{df}{dx} \right|_{x=x_0}} \quad (f(x_0) = 0) \quad ,$$

it can be shown that (cfr  $\sum_X A(X) \delta(\mathcal{H}(X) - E)$  )

$$\begin{aligned} \frac{1}{N!} \int dp_s \int ds \int d\vec{P} \int d\vec{R} \, A\left(\frac{\vec{P}}{s}, \vec{R}\right) \delta\left(E - \mathcal{H}\left(\vec{P}, \vec{R}, p_s, s\right)\right) \\ = \frac{1}{3N+1} \sqrt{\frac{2\pi Q}{k_B T}} \exp \frac{E}{k_B T} \\ \times \frac{1}{N!} \int d\vec{P}' \int d\vec{R}' A(\vec{P}', \vec{R}') \exp -\frac{\mathcal{H}_0\left(\vec{P}', \vec{R}'\right)}{k_B T} \end{aligned}$$

- RESCALING:  $\left(\vec{R}' = \vec{R}, \vec{P}' = \frac{\vec{P}}{s}, dt' = \frac{dt}{s(t)}\right)$

## 8.5.1 The Nosé-Hoover thermostat (IV)

- One performs the simulation with virtual variables  $(\vec{R}, \vec{P}, p_s, s, t)$ .  
The real (or, physical) variables are  
$$\left( \vec{R}'(t) = \vec{R}(t), \vec{P}'(t) = \frac{\vec{P}(t)}{s(t)}, dt' = \frac{dt}{s(t)} \right).$$
- Simulation of the system  $\mathcal{H}(\vec{R}, \vec{P}, p_s, s)$  at constant energy conditions is like running  $\mathcal{H}_0(\vec{R}', \vec{P}')$  under constant temperature conditions!
- Expectation value of a physical variable  $A(\vec{R}', \vec{P}')$

$$\left\langle A\left(\vec{R}, \frac{\vec{P}}{s}\right) \right\rangle = \left\langle A\left(\vec{R}', \vec{P}'\right) \right\rangle_c$$

where  $\langle \dots \rangle_c$  denotes average in the canonical ensemble and  $\langle \dots \rangle$  can be computed under constant-energy conditions (microcanonical ensemble)!

## 8.5.1 The Nosé-Hoover thermostat (V)

- **The equations of motion in the virtual variables** (derived by simply using the Hamilton's equations)

$$\frac{d^2\vec{r}_i(t)}{dt^2} = -\frac{1}{ms^2(t)}\vec{\nabla}_i \sum_{i < j} U(\vec{r}_i - \vec{r}_j) - \frac{2}{s(t)} \frac{d\vec{r}_i(t)}{dt} \frac{ds(t)}{dt}.$$

- ordinary force has an extra factor  $\frac{1}{s^2(t)}$
- additional frictional force with strength  $\frac{2}{s(t)} \frac{ds}{dt}$
- Equations of motion for the real variables  $(\vec{R}', \vec{P}', t')$  can be easily derived from those in the variables  $(\vec{R}, \vec{P}, s, p_s, t)$ !

## 8.5.1 The Nosé-Hoover thermostat (VI)

ADDITIONAL COMPLICATION in the calculation of averages:

- relation between virtual time ( $t$ ) and real time ( $t'$ )

$$dt' = \frac{dt}{s(t)}$$

- The real time steps are NOT equidistant and time averaging in real time is therefore not equivalent to averaging in virtual time!
- Can be solved by computing the following averages in the simulations

$$\frac{\left\langle A\left(\vec{R}, \frac{\vec{P}}{s}\right) \right\rangle}{\left\langle \frac{1}{s} \right\rangle}$$

and putting  $g = 3N$  instead of  $g = 3N + 1$ .

## 8.5.2 Keeping the pressure $p$ constant (I)

- Partition function in  $(N, p, T)$  ensemble is given by

$$Q(N, p, T) = \frac{1}{N!} \frac{1}{h^{3N}} \int dV \exp -\beta pV \int d\vec{R} d\vec{P} \exp -\beta \mathcal{H}(\vec{R}, \vec{P}) .$$

- Simulation is performed with a Hamiltonian consisting of  $6N + 4$  virtual variables **(BOTH THE “HEAT BATH” AND THE “PISTON” ARE INCLUDED AS DEGREES OF FREEDOM)**

$$\begin{aligned} \mathcal{H}(\vec{R}, \vec{P}, p_s, s, p_V, V) &= \sum_{i=1}^{i=N} \frac{\vec{p}_i^2}{2ms^2V^{2/3}} + \sum_{i < j} U(V^{1/3} |\vec{r}_i - \vec{r}_j|) \\ &+ \frac{p_s^2}{2Q} + (3N + 1) k_B T \ln s + pV + \frac{p_V^2}{2W} . \end{aligned}$$

The Hamiltonian has an extra potential energy ( $pV$ ) and an extra kinetic energy ( $p_V^2/2W$ , with  $W$  the mass of the piston).

## 8.5.2 Keeping the pressure $p$ constant (II)

- One performs the simulation with virtual variables  $(\vec{R}(t), \vec{P}(t), p_s(t), s(t), t)$  under constant  $(E, V, N)$  conditions with the Hamiltonian  $\mathcal{H}(\vec{R}, \vec{P}, p_s, s, p_V, V)$
- The real (or, physical) variables are related to the virtual variables

$$\left( \vec{R}'(t) = \vec{R}(t) V^{\frac{1}{3}}(t) \quad ; \quad \vec{P}'(t) = \frac{\vec{P}(t)}{s(t) V^{\frac{1}{3}}(t)} \quad ; \quad dt' = \frac{dt}{s(t)} \right)$$

## 8.5.2 Keeping the pressure $p$ constant (III)

- in the simulation (microcanonical or  $(E, V, N)$  conditions!) one can determine the expectation values

$$\left\langle A \left( \vec{R}(t) V^{\frac{1}{3}}(t), \frac{\vec{P}(t)}{s(t) V^{\frac{1}{3}}} \right) \right\rangle$$

- these expectation values are identical to what one obtains under  $(N, p, T)$  conditions for a Hamiltonian of the type

$$\mathcal{H}_0 \left( \vec{P}', \vec{R}' \right) = \sum_{i=1}^{i=N} \frac{(\vec{p}'_i)^2}{2m} + \sum_{i < j} U \left( \vec{r}'_i - \vec{r}'_j \right)$$

- the distribution of configurations in an  $(N, p, T)$  ensemble

$$\rho \left( \vec{P}', \vec{R}', V \right) = V^N \exp - \frac{1}{k_B T} \left[ \mathcal{H}_0 \left( \vec{P}', \vec{R}' \right) + pV \right]$$

## 8.5.2 Keeping the pressure $p$ constant (IV)

The equations of motion of the Hamiltonian  $\mathcal{H}(\vec{R}, \vec{P}, p_s, s, p_V, V)$

232

*Molecular dynamics simulations*

The equations of motion now read:

$$\frac{d\mathbf{r}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}_i}{mV^{2/3}s^2} \quad (8.101a)$$

$$\frac{ds}{dt} = \frac{\partial \mathcal{H}}{\partial p_s} = \frac{p_s}{Q} \quad (8.101b)$$

$$\frac{d\mathbf{p}_i}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{r}_i} = -\nabla_i U(V^{1/3}R) \quad (8.101c)$$

$$\frac{dp_s}{dt} = -\frac{\partial \mathcal{H}}{\partial s} = \left( \frac{\sum_i p_i^2}{mV^{2/3}s^2} - gk_B T \right) / s \quad (8.101d)$$

$$\frac{dV}{dt} = \frac{\partial \mathcal{H}}{\partial p_V} = \frac{p_V}{W} \quad (8.101e)$$

$$\frac{dp_V}{dt} = -\frac{\partial \mathcal{H}}{\partial V} = \left( \frac{\sum_i p_i^2}{mV^{2/3}s^2} - \sum_i \nabla_i U(V^{1/3}R) \cdot \mathbf{r}_i \right) / (3V) - p. \quad (8.101f)$$

## 8.8 Langevin dynamics simulation (I)

- MODEL: highlight the interesting features and approximate the less relevant parts
- Langevin dynamics technique: solution with particles that are much heavier than the solvent molecules
  - 1 kinetic energy is on average divided equally over the degrees of freedom: heavier particles move slowly and change their momenta after many collisions with solvent molecules
  - 2 heavier particles have a larger time scale than the solvent molecules
  - 3 effect of the solvent molecules on the heavier particles can be treated approximately by means of the combination of a friction force and a random force
- Langevin equation and dynamics on WIKI
  - 1 [http://en.wikipedia.org/wiki/Langevin\\_equation](http://en.wikipedia.org/wiki/Langevin_equation)
  - 2 [http://en.wikipedia.org/wiki/Langevin\\_dynamics](http://en.wikipedia.org/wiki/Langevin_dynamics)

## 8.8 Langevin dynamics simulation (II)

### Langevin equation

Equation of motion for the heavy particles (example of a Stochastic Differential Equation (SDE) )

$$m \frac{d\vec{v}}{dt} = -\gamma \vec{v}(t) + \vec{F}(t) + \vec{R}(t)$$

- $-\gamma \vec{v}(t)$  friction force: coarse grained (or, averaged) effect of the collisions between heavy particles and the solvent molecules (more encounters at the front than at the back)
- $\vec{F}(t)$  external force: gravitation, walls, etc
- $\vec{R}(t)$  random force: frequent collisions with the solvent particles (fluctuations on top of the friction force)

[http://en.wikipedia.org/wiki/Stochastic\\_differential\\_equation](http://en.wikipedia.org/wiki/Stochastic_differential_equation)

## 8.8 Langevin dynamics simulation (III)

- fluid particles show time correlation effects that are poorly accounted for in the Langevin equation
- can be accounted for by replacing the instantaneous velocity by a memory kernel that includes the velocities at previous times

$$-\gamma v(t) \rightarrow - \int_{-\infty}^t dt' \gamma(t-t') v(t')$$

- approach with memory kernels has resulted in a whole industry of so-called **Generalised Langevin-dynamics Simulations**
- another improvement: interaction between heavy particles will affect the friction term (**Dissipative Particle Dynamics**)

## 8.8 Langevin dynamics simulation (IV)

### Random force in Langevin dynamics

- Average effect is absorbed in friction term

$$\langle R(t) \rangle = 0$$

- Values of  $R(t)$  are uncorrelated

$$\langle R(t)R(t + \tau) \rangle = 0 \quad \text{for, } \tau > 0$$

- Values of  $R(t)$  are distributed according to a Gaussian

$$P[R(t)] = \frac{1}{\sqrt{2\pi \langle R^2 \rangle}} \exp -\frac{R^2(t)}{2 \langle R^2 \rangle}$$

## 8.8 Langevin dynamics simulation (V)

- All the assumptions of previous slide can be summarized in the following prescription: Probability for a set of random forces to occur between  $t_0$  and  $t_1$

$$P[R_i(t)]|_{t_0 < t < t_1} \sim \exp -\frac{1}{2q} \int_{t_0}^{t_1} dt R_i^2(t)$$

with  $q$  a constant that can be associated with the temperature of the system!

- the correlation function reads

$$\langle R(t)R(t+\tau) \rangle = q\delta(\tau)$$

## 8.8 Langevin dynamics simulation (VI)

For  $\vec{F}(t) = 0$  the Langevin equation has the following analytical solution

$$\vec{v}(t) = \vec{v}(0) \exp \frac{-\gamma t}{m} + \frac{1}{m} \int_{\tau=0}^{\tau=t} d\tau \vec{R}(\tau) \exp \frac{-\gamma(t-\tau)}{m}$$

To prove this one can make use of the fact that

$$\begin{aligned} \frac{\partial}{\partial \alpha} \int_{x=f(\alpha)}^{x=g(\alpha)} dx F(\alpha, x) &= \int_{x=f(\alpha)}^{x=g(\alpha)} \frac{\partial F(\alpha, x)}{\partial \alpha} dx \\ &+ \left[ \frac{\partial g(\alpha)}{\partial \alpha} F(\alpha, g(\alpha)) - \frac{\partial f(\alpha)}{\partial \alpha} F(\alpha, f(\alpha)) \right], \end{aligned}$$

## 8.8 Langevin dynamics simulation (VII)

Remember that:  $\langle R(t) \rangle = 0$  and  $\langle R(t)R(t+\tau) \rangle = q\delta(\tau)$

- **the expectation value of the velocity at some time  $t$**

$$\langle v(t) \rangle = v(0) \exp \frac{-\gamma t}{m}$$

velocity of a particle subject to a frictional force (no net effect from  $R(t)$ )

- **the expectation value of the (velocity)<sup>2</sup> at some time  $t$**

$$\langle v^2(t) \rangle = v^2(0) \exp \frac{-2\gamma t}{m} + \frac{q}{2\gamma m} \left( 1 - \exp \frac{-2\gamma t}{m} \right)$$

- **the velocity autocorrelation function (VACF)**

$$\langle v(t)v(0) \rangle = v^2(0) \exp \frac{-\gamma t}{m}$$

Oversimplification: misses the long time tail typical for fluids!

## 8.8 Langevin dynamics simulation (VIII)

**For large  $t$ : velocity is distributed according to a Gaussian**

- 1 for sufficiently large  $t$  one has

$$\langle v(t) \rangle \rightarrow 0 \quad \text{and,} \quad \langle v^2(t) \rangle \rightarrow \frac{q}{2\gamma m}$$

- 2  $v(t)$  depends linearly on the random force (which is distributed according to a Gaussian)
- 3 from the above considerations: for sufficiently large  $t$  the  $v$  is distributed according to a Gaussian with zero mean and  $\sigma^2 = \langle v^2 \rangle$

$$P[v(t)] = \sqrt{\frac{\gamma m}{\pi q}} \exp -\frac{v^2 \gamma m}{q}$$

This is a Maxwell distribution for  $q = 2k_B T \gamma$  (one can determine  $q$  so as to reach a desired temperature  $T$ )

## 8.8 Langevin dynamics simulation (IX)

### Algorithm for simple Langevin dynamics

Taylor expansion for  $x(h)$  and  $x(-h)$  results in

$$x(h) \left[ 1 + \frac{\gamma h}{2m} \right] + x(-h) \left[ 1 - \frac{\gamma h}{2m} \right] = 2x(0) + h^2 \left[ \frac{F(0)}{m} + \frac{R_+ + R_-}{2m} \right]$$

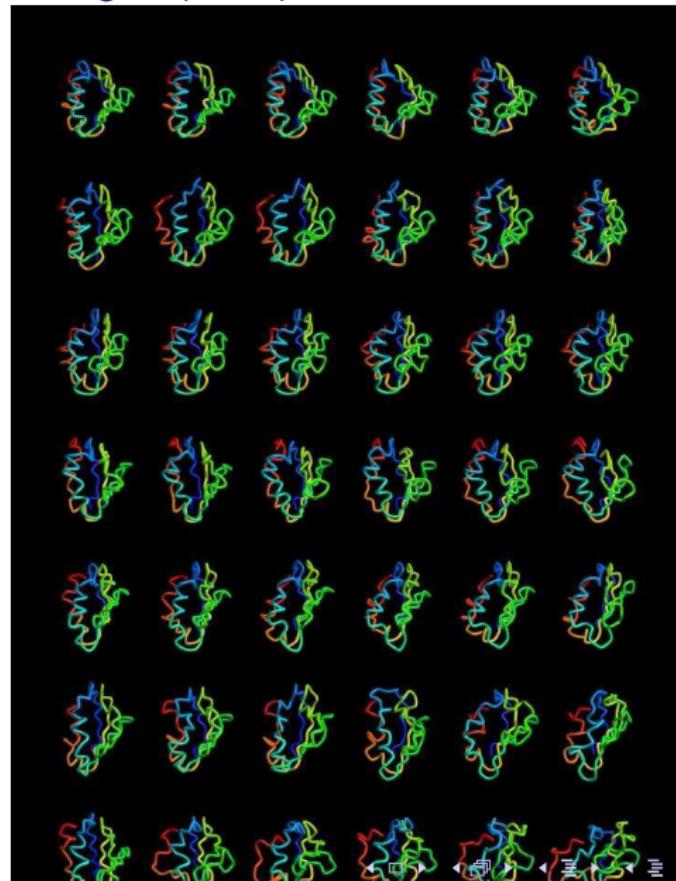
with  $R_+$  ( $R_-$ ) the random force during the time interval  $[0, h]$  ( $[-h, 0]$ )

- at each step a new value for  $R_+$  must be drawn from a Gaussian random generator
- an ideal time step  $h$  is impossible to realize
  - 1  $h$  is determined by the requirement that  $\vec{F}$  remains constant between  $t$  and  $t + h$
  - 2 random force is not as “static” as  $\vec{F}$ , requires smaller time steps

## 8.8 Langevin dynamics simulation X

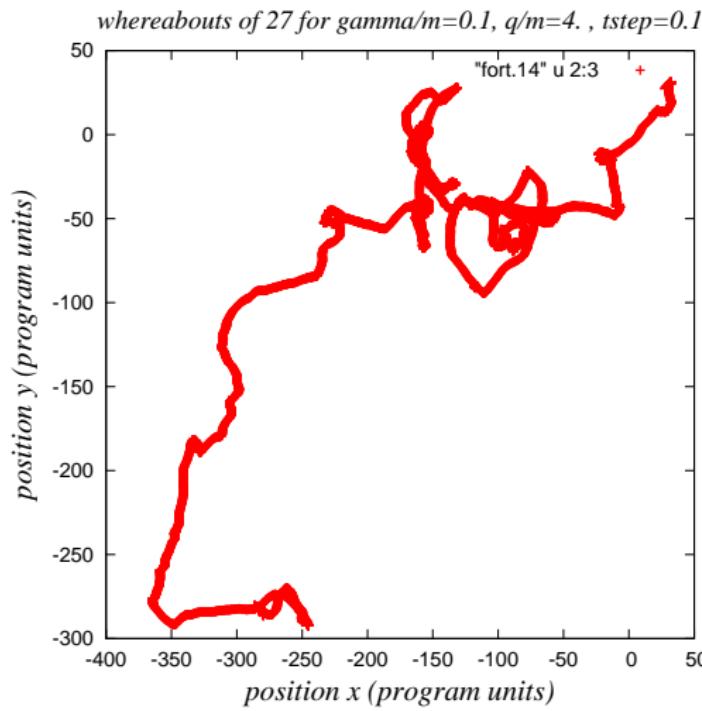
Molecular Dynamics in Groningen (RUG)

Snapshots taken at various times during 200 nanosecond Langevin Dynamics simulations of Hpr for seven different temperatures. (bottom row is hottest)



## 8.8 Langevin dynamics simulation (XI)

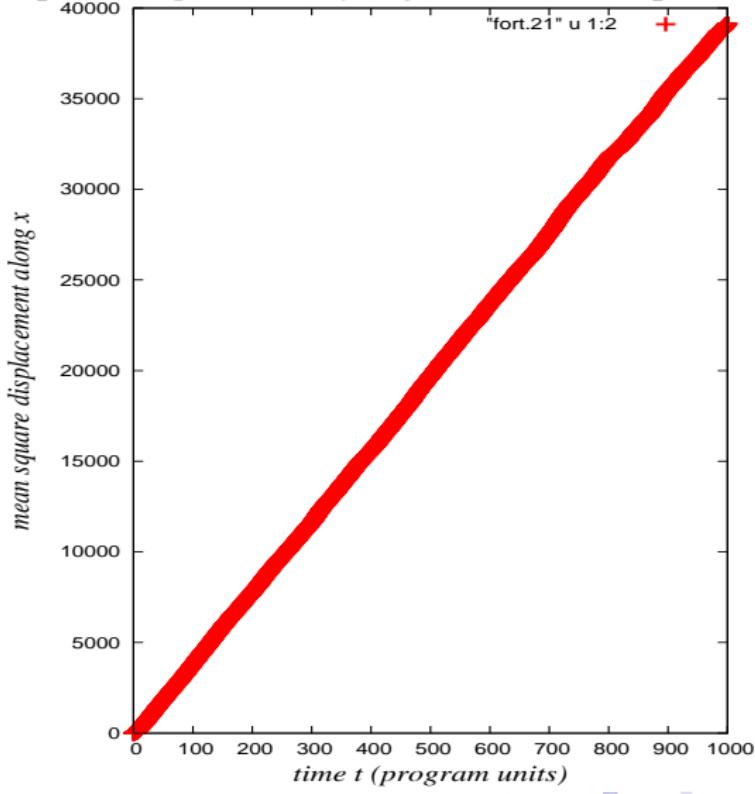
Example: simulation with 10000 particles over 10000 time steps in two dimensions.



The motion of each individual particle looks like a random walk

## 8.8 Langevin dynamics simulation (XII)

mean square displacement for  $\gamma/m=0.1$ ,  $q/m=4.$  ,  $tstep=0.1$



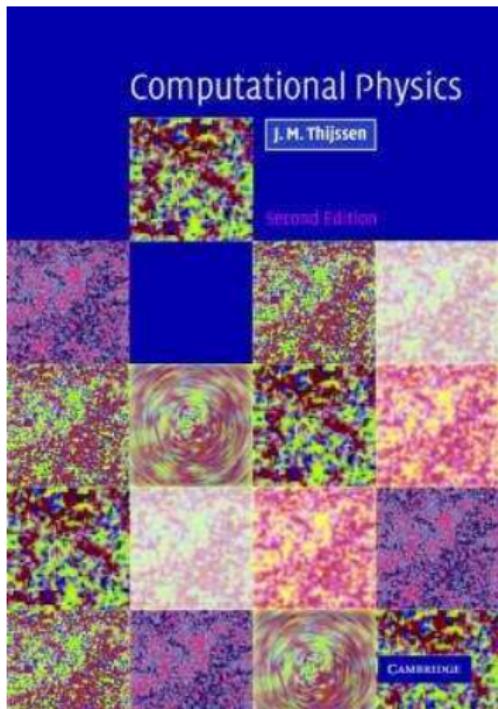
The diffusion is Gaussian  
 $\langle \Delta x^2(t) \rangle \sim t$

# COMPUTATIONAL PHYSICS/ CHAPTER 10

J. Ryckebusch

Ghent University

Academic Year 2013-2014



J.M. Thijssen  
“Computational Physics”  
(1999)

- Chapter 1: Introduction (p1-13)
- Chapter 2: Quantum scattering with a spherically symmetric potential
- Chapter 3: The variational method for the Schrödinger equation
- Chapter 4: The Hartree-Fock method
- Chapter 5: Density functional theory
- Chapter 7: Classical statistical physics
- Chapter 8: Molecular dynamics simulations
- Chapter 9: Quantum molecular dynamics
- Chapter 10: The Monte-Carlo method (p295-337)

# CHAPTER 10: The Monte Carlo Method

10.1 Introduction

10.2 Monte Carlo integration

10.3 Importance sampling through Markov chains

10.3.1 Monte Carlo for the Ising model

10.3.2 Monte Carlo simulation of a monoatomic gas

10.4 Other ensembles

10.4.1 The ( $NPT$ ) ensemble

10.4.2 The grand canonical ensemble

App A A.6 Numerical quadratures (p566-568)

## 10.1 Introduction (p295)

$$\bar{A} = \frac{\sum_x A(x) \delta(\mathcal{H}(x) - E)}{\sum_x \delta(\mathcal{H}(x) - E)}$$

- 1 MD method: statistical averages of static and dynamic physical quantities of classical many-particle systems by solving equations-of motion
- 2 MC method: simulating classical many-particle systems by introducing artificial dynamics based on random numbers



## 10.1 Introduction (2)

Monte Carlo (MC) method: a numerical technique in which random numbers play an essential role

- 1 **Direct MC:** random numbers mimic the effect of complicated processes, the details of which are not crucial
  - modelling traffic: [http://en.wikipedia.org/wiki/Traffic\\_flow](http://en.wikipedia.org/wiki/Traffic_flow)
  - Langevin dynamics (random forces) to model diffusion properties
  - Financial markets
- 2 **MC integration:** method for computing integrals over high-dimensional volumes
- 3 **Metropolis MC:** sequence of distributions of a system in a so-called Markov chain. This method allows to study the properties of classical and quantum many-particle systems (Chapter 12).

## Appendix A.6 Numerical quadratures (p566)

- Numerical integration (or, quadrature) of a well-behaved function  $f(x)$  on the interval  $[a, b]$  can be done straightforwardly in an equidistant grid

$$x_n = a + nh \quad h = \frac{b - a}{N} \quad n = 0, 1, 2, \dots, N$$

- Crude approximation

$$\int_{a=x_0}^{b=x_N} f(x) dx \approx h \sum_{n=0}^{N-1} f(x_n) + \mathcal{O}(h)$$

$\mathcal{O}(h)$ : Doubling the number of integration points reduces the error  $\mathcal{O}$  by about a factor of two (since  $f(x_{n+1}) - f(x_n)$  is of the order  $h$ )

## Appendix A.6 Numerical quadratures (2)

- Above can be improved by approximating  $f(x)$  in  $[x_n, x_{n+1}]$  (Taylor expansion)

$$f(x) \approx f(x_n) + \frac{(x - x_n)}{h} [f(x_{n+1}) - f(x_n)] + \mathcal{O}(h^2)$$

The function  $f(x)$  is approximated by a polynomial of first order constructed with  $f(x_n)$  and  $f(x_{n+1})$

- Integrating this form analytically over the complete interval leads to

$$\begin{aligned} \int_a^b f(x) dx &= h \left[ \frac{1}{2} f(x_0) + f(x_1) + f(x_2) + \dots \right. \\ &\quad \left. + f(x_{N-1}) + \frac{1}{2} f(x_N) \right] + \mathcal{O}(h^2) \end{aligned}$$

**trapezoidal rule: quadrature which has an accuracy up to  $\mathcal{O}(h^2)$**

## Appendix A.6 Numerical quadratures (3)

- One can obtain a quadrature with a higher accuracy by working on two slices together!
- Using the technique of “Lagrange polynomial interpolation” one can construct a second-order polynomial which approximates  $f(x)$  in the interval  $[a, b]$ . This requires three points:  $f(a), f\left(\frac{a+b}{2}\right), f(b)$

$$\begin{aligned} f(x) &\approx \frac{(x-a)\left(x-\frac{a+b}{2}\right)}{(b-a)\left(b-\frac{a+b}{2}\right)} f(b) \\ &+ \frac{\left(x-\frac{a+b}{2}\right)(x-b)}{\left(a-\frac{a+b}{2}\right)(a-b)} f(a) \\ &+ \frac{(x-b)(x-a)}{\left(\frac{a+b}{2}-b\right)\left(\frac{a+b}{2}-a\right)} f\left(\frac{a+b}{2}\right) + \mathcal{O}(h^3) \end{aligned}$$

$f(x)$  is a polynomial of order two!

## Appendix A.6 Numerical quadratures (4)

- With the above second-order polynomial approximation to  $f(x)$  one can compute the integral

$$\int_a^b f(x) dx = \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] + \mathcal{O}((b-a)^5)$$

- In a next step one can split up  $[a, b]$  in  $\frac{N}{2}$  subintervals with size  $2h$  to obtain

$$\begin{aligned} \int_a^b f(x) dx &= \frac{h}{3} \left[ f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots \right. \\ &\quad \left. + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N) \right] + \mathcal{O}(h^4) \end{aligned}$$

**Simpson's rule: quadrature which has an accuracy up to  $\mathcal{O}(h^4)$**

- Method boils down to making a polynomial approximation to integrand and then integrating the formula analytically!

## Appendix A.6 Numerical quadratures (5)

- procedure can be extended: the accuracy is related to the number of points used in fitting the polynomial: for  $n$  points one can determine a polynomial of order  $n - 1$  having the same value as  $f(x)$  in  $n$  points
- tricks are available for integrating functions containing singularities
- Romberg method: consists of three steps
  - 1 Repeatedly use the trapezoidal rule for successive values of  $h$  and determine the integral  $I(h)$

$$h \longrightarrow \frac{h}{2} \longrightarrow \frac{h}{4} \longrightarrow \frac{h}{8} \longrightarrow \dots$$

- 2 Perform a polynomial fit to  $I(h)$
- 3 The  $I(h = 0)$  is very good approximation to the exact value

## Appendix A.6 Numerical quadratures (6)

- The abovementioned methods are based on polynomial fits to the function in  $[a, b]$  and equidistant points.
- Alternative method: Gauss-Legendre method:  
[http://en.wikipedia.org/wiki/Gaussian\\_quadrature](http://en.wikipedia.org/wiki/Gaussian_quadrature)
  - 1 the grid points are determined as the zeroes of the Legendre polynomials
  - 2 in  $[a, b]$  the function  $f(x)$  is approximated by Legendre polynomials
  - 3 Advantage: for given  $N$  better accuracy than with equidistant methods
  - 4  $N$ -point Gauss-Legendre has accuracy of  $2N$  equidistant-point method
  - 5 Resulting algorithm (no derivations):

$$\int_{-1}^{+1} dx f(x) = \sum_{n=1}^N w_n f(x_n) + \mathcal{O}(h^{2N})$$

Here,  $x_n$  are the zeroes of the Legendre polynomial  $P_N$  and  $h = \frac{2}{N}$

## 10.2 Monte Carlo integration (1) (p296)

We want to compute the integral  $I$  of a smooth function  $f(x)$  between  $a$  and  $b$

- Standard numerical methods boil down to computing sums of the type

$$I = \frac{(b-a)}{N} \sum_{i=1}^N w_i f(x_i)$$

- 1 the weights  $w_i$  do not depend on function  $f(x)$
- 2 methods are based on polynomial expansions
- 3 accuracy is of the order  $\sigma \sim h^k \sim N^{-k}$  ( $k$  is a positive integer)
- MC integration:  $w_i = 1$  and  $x_i$  are chosen randomly!

## 10.2 Monte Carlo integration (2)

- If the random coordinates  $x_i$  are homogeneously distributed on  $[a, b]$  and  $N$  is sufficiently large, MC integration will lead to a result close to the exact integral
- We calculate the variance of the result

$$\sigma^2(I) = \left\langle \left( \frac{b-a}{N} \sum_{i=1}^N f(x_i) \right)^2 \right\rangle - \left( \left\langle \frac{b-a}{N} \sum_{i=1}^N f(x_i) \right\rangle \right)^2$$

The  $\langle \dots \rangle$  denote an average over all possible realisations of the sequence of random coordinates  $x_i \in [0, 1]$ .

- The variance can be rewritten as:

$$\sigma^2(I) = \frac{(b-a)^2}{N} \left( \bar{f}^2 - \bar{f}^2 \right)$$

- 1  $\bar{f}$  denotes the average of the function  $f(x)$  on  $[0, 1]$
- 2 reliable error estimate requires that  $f$  is square integrable

## 10.2 Monte Carlo integration (3)

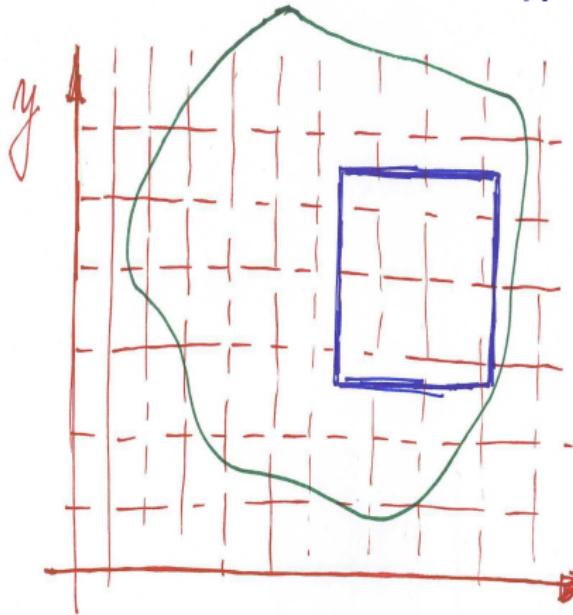
### About the accuracy ( $\sim \sigma$ ) of MC integration

- accuracy  $\sigma(I)$  is proportional to  $1/\sqrt{N}$  (central limit theorem)
- in one dimension: MC integration is less efficient than standard quadrature techniques which reach an accuracy of the order  $\sigma \sim N^{-k}$ ,  $k = 1, 2, \dots$
- in higher dimensions: MC integration is more efficient
  - 1 **Standard technique** with  $\mathcal{O}(h^k)$  in hypercube of dimension  $d$ : in each dimension  $N^{1/d}$  points ; accordingly the error scales like  $\sigma \sim N^{-k/d}$
  - 2 **MC technique**: error is independent of dimension  $d$  and  $\sigma \sim N^{-1/2}$

**MC integration is more efficient than an order- $k$  algorithm when  $d > 2k$**

*Quality of integration is related to the question of how well one can distribute points over a (multi-dimensional) volume!!*

## 10.2 Monte Carlo integration (4)



- counter-intuitive result: in higher dimensions the random distribution is more homogeneous than the regular grid
- homogeneous distribution: number of points within rectangular volume should be proportional to its volume
- with standard regular-grid techniques: number of points it contains grows stepwise as the volume increases
- with random distributions these steps X are extremely unlikely to occur

## 10.2 Monte Carlo integration (5)

Methods to improve the accuracy of the MC integration method

- **Stratified MC** : method to distribute the points more homogeneously by subdividing the integration hypercube into smaller equally sized subvolumes and choose an equal amount of random points in each subvolume
- A large variance of  $f(x)$  over the volume will result in big errors! This contribution to the error can be reduced in so-called **Importance sampling MC**. It consists of introducing a function  $\rho$  which has the following properties

$$\frac{f}{\rho} \approx 1 \text{ in the integration volume}$$

$$\int_V dV \rho = 1$$

## 10.2 Monte Carlo integration (6)

### The importance sampling MC method

- requires a priori knowledge of the function  $f$
- importance sample MC method: a variance reducing technique
- the random points  $x_i$  are chosen from the distribution  $\rho(x)$  (for example with the Von Neumann method of Appendix B3)
- by doing this the error in the result can be reduced considerably

$$\begin{aligned}\sigma^2 &= \frac{1}{N} \left[ \int_a^b \left[ \frac{f(x)}{\rho(x)} \right]^2 \rho(x) dx - \left( \int_a^b \left[ \frac{f(x)}{\rho(x)} \right] \rho(x) dx \right)^2 \right] \\ &< \frac{1}{N} \left( \bar{f}^2 - \bar{f}^2 \right)\end{aligned}$$

- disadvantage: dangerous to choose functions  $\rho(x)$  which approach zero!

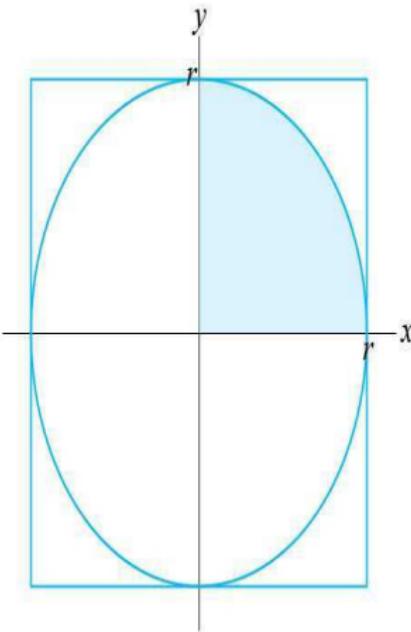
## 10.2 Monte Carlo integration (7)

### The importance sampling MC method

- **Adaptive MC** methods aim at concentrating the sampling points in those regions where  $f$  contributes significantly to the integral.
- adaptive MC locates these regions by probing the function at random points and require no a priori knowledge on the function  $f$  (*adaptive MC learns about the function as it proceeds*)
- MC integration is not susceptible to correlations in the random number generator (as long as it is a good one) and it is possible to generate artificial number sequences which fill a high-dimensional volume very homogeneously (“quasi-Monte Carlo”)

## 10.2 Monte Carlo integration (8)

### Buffon's needle technique to estimate $\pi$ (I)



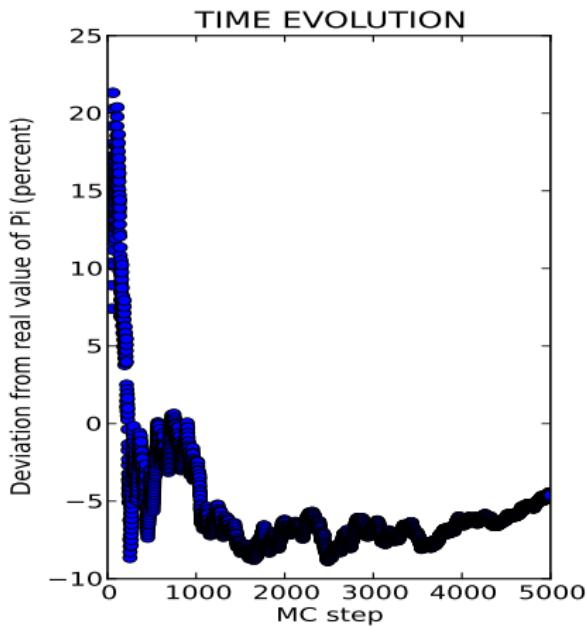
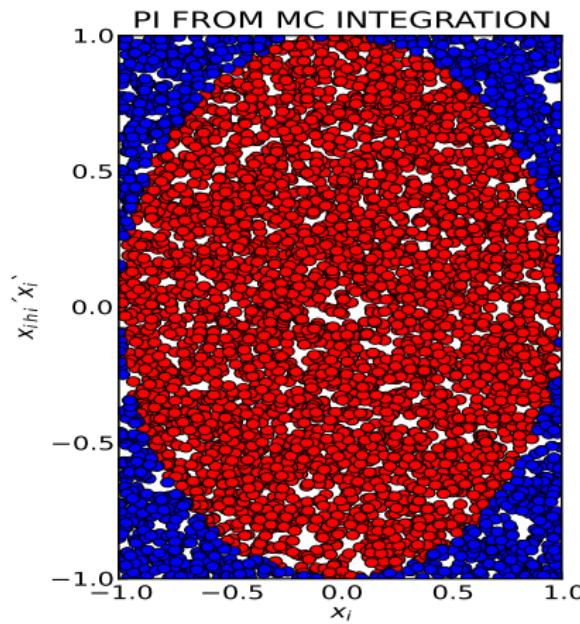
- generate a large number ( $n$ ) of random points inside a square.
- a large number of points ( $m < n$ ) will also fall inside the circle
- by approximation:  $n$  is the area of the square, while  $m$  is the area of the circle
- Calculating these areas we can see that:

$$\frac{m}{n} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4}$$

- By generating a number of random points and checking how many of them fall inside the circle, we can approximate  $\pi$
- Python implementation: COMPUTEPI.PY

## 10.2 Monte Carlo integration (9)

### Buffon's needle technique to estimate $\pi$ (II)



Result from the Python implementation COMPUTEPI.PY which illustrates  $\frac{1}{\sqrt{n}}$  accuracy of the estimate!

## 10.3 Importance sampling through Markov chains (p275)

### Importance sampling method for classical many-particle systems in the canonical or ( $NVT$ ) ensemble

- Phase space (e.g.  $\int d\vec{R} \int d\vec{P}$ ) is highly dimensional
- The configurations  $X$  (e.g.  $\vec{R}\vec{P}$ ) should be weighted according to the Boltzmann factor

$$\rho(X) = \exp[-\beta E(X)] \quad \text{with,} \quad E(X) \geq 0$$

- When generating random configurations and accepting them with probability  $\exp[-\beta E(X)]$  one runs into difficulties.
- Indeed, the number of configurations with a particular energy increases very strongly with energy ( $\sim E^{\text{number of d.o.f.}}$ ).
- Hence, for finite temperature most of the randomly generated configurations have a high energy ...
- Most of the simulation time is used to generate configurations which are rejected on the basis of the Boltzmann factor

( ! ! HIGHLY INEFFICIENT ! ! )

## 10.3 Importance sampling through Markov chains (2)

- **QUESTION:** How to generate statistically independent configurations with a bias towards lower energies (dictated by the Boltzmann weight)?

**ANSWER:** Algorithms exist but are very time consuming

- **Metropolis method (1953)** abandons the idea of constructing statistically independent configurations (*configurations are constructed in a so-called Markov chain: each configuration is generated with a probability distribution depending on the previous configuration*)

$$P_N(X_1, X_2, \dots, X_N) = P_1(X_1) T(X_1 \rightarrow X_2) T(X_2 \rightarrow X_3) \dots T(X_{N-1} \rightarrow X_N)$$

- 1 With (normalisation of the transition probabilities)

$$\sum_{X'} T(X \rightarrow X') = 1.$$

- 2 Accordingly:  $\sum_{X_1 X_2 \dots X_N} P_N(X_1, X_2, \dots, X_N) = 1$

## 10.3 Importance sampling through Markov chains (2)

- QUESTION: How to generate statistically independent configurations with a bias towards lower energies (dictated by the Boltzmann weight)?

ANSWER: Algorithms exist but are very time consuming

- **Metropolis method (1953)** abandons the idea of constructing statistically independent configurations (*configurations are constructed in a so-called Markov chain: each configuration is generated with a probability distribution depending on the previous configuration*)

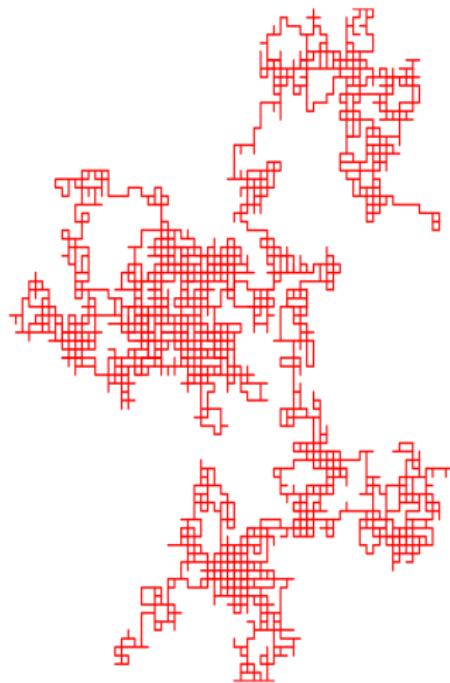
$$P_N(X_1, X_2, \dots, X_N) = P_1(X_1) T(X_1 \rightarrow X_2) T(X_2 \rightarrow X_3) \dots T(X_{N-1} \rightarrow X_N)$$

- 1 With (normalisation of the transition probabilities)  
 $\sum_{X'} T(X \rightarrow X') = 1.$
- 2 Accordingly:  $\sum_{X_1 X_2 \dots X_N} P_N(X_1, X_2, \dots, X_N) = 1$

## 10.3 Importance sampling through Markov chains (3)

### Examples of Markovian and nonMarkovian sequences

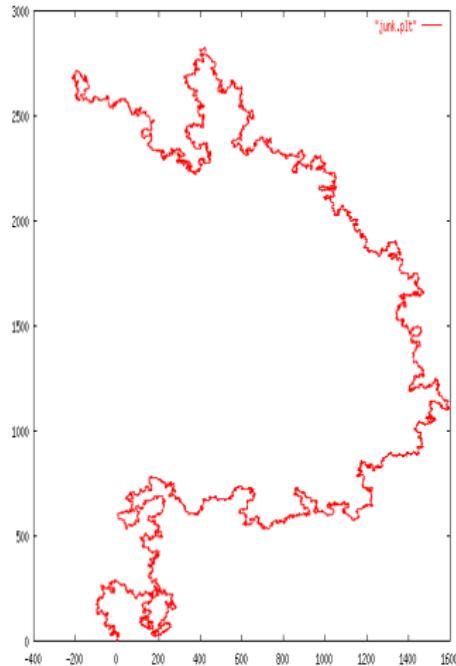
- Markovian sequence: random walk on a two-dimensional square lattice  
*(probability to jump to each of its nearest neighbours is 1/4 - independent of the history of the walker)*
- nonMarkovian (or, correlated) sequence: the self-avoiding random walk (walker is not allowed to visit a site which has been visited in the past - current probability depends on the full history of the walk)



## 10.3 Importance sampling through Markov chains (3)

### Examples of Markovian and nonMarkovian sequences

- Markovian sequence: random walk on a two-dimensional square lattice  
*(probability to jump to each of its nearest neighbours is 1/4 - independent of the history of the walker)*
- nonMarkovian (or, correlated) sequence: the self-avoiding random walk (*walker is not allowed to visit a site which has been visited in the past - current probability depends on the full history of the walk*)



## 10.3 Importance sampling through Markov chains (4)

Objective: generate a Markov chain of system configurations such that they have a distribution proportional to  $\exp -\beta E(X)$ . Conditions:

- 1 distribution should be independent of the position in the chain
- 2 distribution should be independent of the initial configuration (it needs some time to “forget” the chosen initial condition)

### Conditions for generating ERGODIC Markov chain

- 1 **Connectedness, irreducibility:** Every configuration should be accessible from every other configuration within finite number of steps
- 2 **NO periodicity:** Periodicity means that it is NOT possible to return to the same configuration except after  $t = nk$  steps ( $n = 1, 2, \dots$  en  $k$  is fixed)

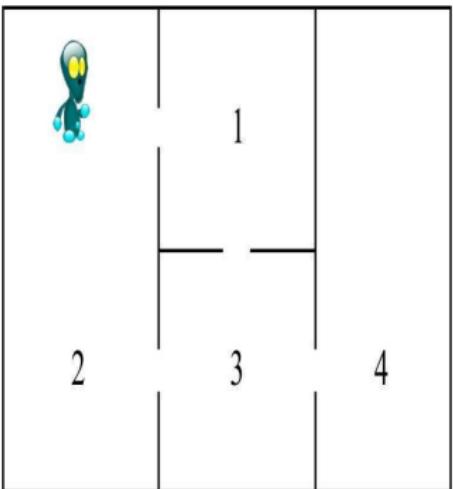
# Example of a Markov process ...

RANDOM WALK OF AN ALIEN IN A HOUSE ...

What fraction of the time will alien spend in each room ???

Steady-state probability to reside in room  $X$ :  
 $\rho(X)$

- $\rho(X') = \sum_X T(X' \leftarrow X) \rho(X)$
- $\rho(X)$  is eigenvector of  $T(X' \leftarrow X)$  associated with the eigenvalue 1
- Transition matrix is determined by:



Transition probability

$T(X' \leftarrow X)$

$X, X' = 1, 2, 3, 4$

$$T(X' \leftarrow X) = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 1 \\ 0 & 0 & \frac{1}{3} & 0 \end{bmatrix}$$

- solution:

$$\rho(1) = \rho(2) = \frac{1}{4}; \rho(3) = \frac{3}{8}; \rho(4) = \frac{1}{8}$$

## 10.3 Importance sampling through Markov chains (5)

**Metropolis MC: generate a Markov chain of configurations with a required invariant distribution**

Find a  $T(X \rightarrow X')$  such that  $\rho(X) = \exp -\beta \mathcal{H}(X)$  (Boltzmann distribution)

- $N$  total number of configurations  $X$
- vector  $\rho(X)$  has  $N$  elements
- matrix  $T(X \rightarrow X')$  has  $N \times N$  elements
- some freedom in finding a solution as there are many different solutions
- referring to the example of the alien: a given  $\rho(X)$  can be an eigenvector of many different matrices  $T(X \rightarrow X')$

## 10.3 Importance sampling through Markov chains (6)

- $\rho(X, t)$ : probability of occurrence of configuration  $X$  at Markov step  $t$
- at large  $t$ :  $\rho(X, t) \rightarrow \rho(X)$  stationary distribution or ergodic Markov chain
- change in  $\rho(X, t)$  at time  $t$  ??
  - 1 decrease in  $\rho(X, t)$ :  $X$  at  $t \rightarrow X'$  at  $t'$
  - 2 increase in  $\rho(X, t)$ :  $X'$  at  $t \rightarrow X$  at  $t'$

### MASTER EQUATION

$$\begin{aligned}\rho(X, t+1) - \rho(X, t) &= - \sum_{X'} T(X \rightarrow X') \rho(X, t) \\ &\quad + \sum_{X'} T(X' \rightarrow X) \rho(X', t)\end{aligned}$$

## 10.3 Importance sampling through Markov chains (7)

- Stationary distribution if  $\rho(X, t+1) = \rho(X, t)$

$$\sum_{X'} T(X \rightarrow X') \rho(X, t) = \sum_{X'} T(X' \rightarrow X) \rho(X', t)$$

- Solution to this equation:

$$T(X \rightarrow X') \rho(X) = T(X' \rightarrow X) \rho(X') \quad \text{DETAILED BALANCE}$$

- Interpretation of the principle of detailed balance:

- 1 associate each  $X$  with a bucket containing  $\rho(X)$  water
- 2 bucket  $X \rightarrow$  bucket  $X'$ : pumping rate  $T(X \rightarrow X') \rho(X)$
- 3 pumping rate between any pair of buckets is balanced:

$$(\text{flow } X \rightarrow X') = (\text{flow } X' \rightarrow X)$$

- 4 obviously: the water volumes  $\rho(X)$  and  $\rho(X')$  do not change

## 10.3 Importance sampling through Markov chains (8)

Make detailed balance solution suitable for practical purposes

Define  $T(X \rightarrow X') \equiv \omega_{XX'} A_{XX'}$  With

1  $\omega_{XX'} = \omega_{X'X}$  ;  $0 \leq \omega_{XX'} \leq 1$  ;  $\sum_{X'} \omega_{XX'} = 1$

TRIAL STEP PROBABILITY

2  $0 \leq A_{XX'} \leq 1$

ACCEPTANCE PROBABILITY

The principle of detailed balance

$$T(X \rightarrow X')\rho(X) = T(X' \rightarrow X)\rho(X')$$

Translates into

$$\frac{A_{XX'}}{A_{X'X}} = \frac{\rho(X')}{\rho(X)}$$

- The detailed balance condition can be fulfilled by many algorithms!!
- The most famous one: the Metropolis method

## 10.3 Importance sampling through Markov chains (9)

### Metropolis Algorithm

Proceeds in three steps:

- STEP1: given  $X$ , suggest  $X'$  with probability  $\omega_{X'X}$
- STEP2: compare  $\rho(X)$  and  $\rho(X')$ 
  - $\rho(X') \geq \rho(X)$ :  $A_{XX'} = 1$
  - $\rho(X') < \rho(X)$ :  $A_{XX'} = \frac{\rho(X')}{\rho(X)}$
- STEP3: acceptance or rejection of  $X'$ 
  - if state  $X'$  is accepted it replaces  $X$
  - if state  $X'$  is NOT accepted system remains in  $X$

## 10.3 Importance sampling through Markov chains (10)

### A few remarks regarding the Metropolis algorithm:

- $X \rightarrow X'$  and  $\rho(X') < \rho(X)$ : we accept the new state with probability  $A_{XX'} = \frac{\rho(X')}{\rho(X)} < 1$  and reject it with probability  $(1 - A_{XX'})$
- can be achieved with following procedure: Generate a random number  $r$  between 0 and 1
  - $r < A_{XX'} : X'$  is accepted
  - $r > A_{XX'} : X'$  is NOT accepted
- (if this procedure is carried out many times state will be accepted a fraction  $A_{XX'}$  of the total number of trials )*
- Markov chain induces correlations
- for sufficiently long times  $\rho(X)$  will reach the required invariant distribution (e.g. the Boltzmann one)

## 10.3 Importance sampling through Markov chains (11)

### A few remarks regarding the Metropolis algorithm: (continued)

- number of statistically independent solutions

$$\frac{\text{total number of steps (in MC steps)}}{\text{correlation time (in MC steps)}}$$

- total number of steps  $\neq$  total number of successful steps
- Metropolis MC generates a sequence of configurations  $X$  with distribution  $\exp -\beta E(X)$ . Ensemble average of a physical quantity

$$\bar{A} = \frac{1}{n - n_0} \sum_{v > n_0}^n A_v$$

$n_0$  number of steps before equilibrium is reached

- $\bar{A}$  is like in MD simulations but  $n$  is NOT physical time

## 10.3 Importance sampling through Markov chains (12)

Detailed balance  $T(X \rightarrow X')\rho(X) = T(X' \rightarrow X)\rho(X')$  is also fulfilled for

- Barker algorithm
- generalised Metropolis method (smart Monte Carlo)
- heat-bath algorithm
- **YOUR ALGORITHM**

### Barker algorithm

$$T(X \rightarrow (X' \neq X)) = \omega_{XX'} \frac{\rho(X')}{\rho(X) + \rho(X')} \quad (\omega_{XX'} = \omega_{X'X})$$
$$T(X \rightarrow X) = 1 - \sum_{X' \neq X} T(X \rightarrow X')$$

## 10.3 Importance sampling through Markov chains (13)

### Generalised Metropolis algorithm

- we assume that for some pairs  $(X, X')$  :  $\omega_{XX'} \neq \omega_{X'X}$
- the acceptance criterion for an attempted move  $X \rightarrow X'$

$$A_{XX'} = \min(1, q_{XX'}) \quad q_{XX'} = \frac{\omega_{X'X} \rho(X')}{\omega_{XX'} \rho(X)}$$

- method is often called: smart Monte Carlo

### Heat-bath algorithm

- trial step involves a few d.o.f. ( $\equiv x$ ), the remaining d.o.f. ( $\equiv (X - x)$ ) are fixed.
- Probability of acceptance

$$\exp[-\beta \mathcal{H}(x | X - x)]$$

$\mathcal{H}(x | X - x)$ : Hamiltonian as function of  $x$  at fixed  $X - x$

- Practical implementation is often difficult ; for lattice models it can be implemented straightforwardly

# Canonical MC method for two examples of classical many-particle systems

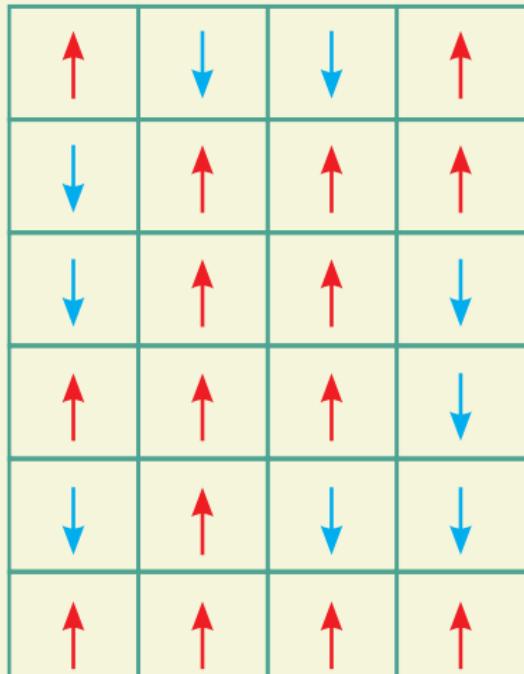
- 10.3.1: Monte Carlo for the Ising model
- 10.3.2: Monte Carlo simulation of a monoatomic gas
- Ising model introduced in Section 7.2.2. (lattice models)
- Ising model described by a Hamiltonian (nearest-neighbor interaction between the spins  $s_i$  in a lattice and interaction of the spins  $s_i$  with an external field  $H$ )

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j - H \sum_i s_i$$

*J: interaction strength of the nearest-neighbor interaction*

- Ising model has NO prescription for its dynamical evolution (MD method requires a differential equation)

## 10.3.1 The Ising model (I)



- A lattice position adopts the value “yes” ( $\uparrow$ ) or “no” ( $\downarrow$ )
- Communication (or, interaction) is restricted to nearest neighbours
- Strength of the communication is a parameter ( $J$ )
- Ferromagnet: the nearest-neighbor interaction attempts to align all spins
- There is a disorder generating component in the system: energy exchange with the environment which gives rise to a non-vanishing temperature
- The magnitude of the disorder generating component is determined by the magnitude of “temperature” ( $T$ )

## The Ising model (II)

- Ising system was originally developed in order to explain spontaneous magnetization
- Ising system is likely the most exported model of physics (applications in neurology, social segregation models, opinion dynamics, behavioral economics, . . ., starting point for lattice calculations in quantum-field theory)
- The laws of Statistical Physics dictate how the Ising system finds its equilibrium between the “order” and “disorder” generating components in the system
- Brute-force solution of the problem is impossible ( $20 \times 20$  two-dimensional lattice has 400 positions which generate  $2^{400}$  different configurations of the system)
- Information about Ising model from computer simulations!

## Predictions from the Ising model (III)

- Ising model displays a phase transition for well-defined values of the ratio

$$\frac{\text{magnitude of the nearest-neighbour interaction}}{\text{magnitude of the temperature}}$$

- Close to the phase transition the physical properties obey power laws
- The coefficients of the power laws are universal
- Close to the phase transition the system is scale invariant (system is identical at long and short-distance scales).
- Scale invariance can be connected to the “Renormalization Group Method (RGM)” (K. Wilson, Nobel Price in Physics, 1982).
- RGM is a mathematical translation of the existence of a hierarchy in a system (systems consist of subsystems, which consist of sub-subsystems, ...) and develops methods to connect all these layers covering very different distance scales!

## 10.3.1 MC for the Ising model (1)

Consider a two-dimensional  $L \times L$  lattice. We have to define

$$T(X \rightarrow X') = \omega_{XX'} A_{XX'}$$

- Choices made for  $\omega_{XX'}$

- $\omega_{XX'} = \frac{1}{L^2}$  if  $X$  and  $X'$  differ by one spin
- $\omega_{XX'} = 0$  otherwise

- With this assumption the Markov chain can be generated

- 1 select a spin at random
- 2 trial configuration ( $= X'$ ): present configuration with selected spin turned over
- 3 compute  $\Delta E(X \rightarrow X') = E(X') - E(X)$ 
  - $\Delta E(X \rightarrow X') < 0$ :  $X'$  is accepted
  - $\Delta E(X \rightarrow X') > 0$ :  $X'$  is accepted with probability  $\exp -\beta \Delta E(X \rightarrow X')$

## 10.3.1 MC for the Ising model (2)

- time in a MC simulation expressed in MCS (= Monte-Carlo steps per spin)  $1\text{MCS} \approx L^2$
- rather than the energy  $E$ , the quantity  $\beta E$  is recorded
- initial state: all the spins are the same (either up or down)

$$\beta E = -2\beta J L^2$$

- energy can be determined in an economical way

$$\begin{array}{ccccccc} X & \rightarrow & X' & \rightarrow & X'' & \rightarrow & \dots \\ E(X) & \rightarrow & E(X') & \rightarrow & E(X'') & \rightarrow & \dots \\ E(X) & \rightarrow & E(X) + \Delta E(X \rightarrow X') & \rightarrow & E(X') + \Delta E(X' \rightarrow X'') & \rightarrow & \dots \end{array}$$

- acceptance probability  $\exp -\beta \Delta E(X \rightarrow X')$  can assume only five different values
- for 2D Ising system: phase transition for  $\frac{J}{k_B T} \approx 0.44$  (*critical fluctuations make it difficult to locate the phase transition*)

## 10.3.1 MC for the Ising model (3)

### Computer code for Metropolis method for two-dimensional Ising model

#### 1 User input

```
0.1 1. 0.02      # Start value, End value and Step value for J/(kB T)
0.0              # Value of H/(kB T)
20              # The linear lattice site (INTEGER!)
120000          # Number of MCS (per site!) (INTEGER!)
```

#### 2 Output of the program:

```
Energy          # MCIter, Energy
Magnet          # MCIter, Magnetisation
SpecificHeat    # J / (kB T), H / (kB T), CVP, CVM, Ener1P, Ener1M
Susceptibility  # J / (kB T), H / (kB T), XIP, XIM, MagnelP, MagnelM
```

- CV: specific heat per lattice site: CVP (CVM) for positive (negative) magnetizations!
- XI: susceptibility per lattice site
- Ener1: energy per lattice site
- Magnel: magnetization per lattice site

## 10.3.1 MC for the Ising model (4)

- for  $T > T_C$  there is no spontaneous magnetization
- for  $T < T_C$  there is spontaneous magnetization which can be difficult to simulate
  - large regions with a “+” spin and large regions with a “-” spin
  - may take long simulation times to have one dominate
  - domain walls separate the “+” regions from the “-” ones and carry a free energy cost  $\Delta F > 0$  per unit length for  $T < T_C$
  - in **reality (=infinite systems)** flipping the magnetization at  $T < T_C$  requires  $\infty$  free energy and does not happen
  - in **simulations (=finite systems)** for  $T < T_C$  magnetization flips around a positive and negative equilibrium value for relatively long times and long-time average vanishes

## 10.3.1 MC for the Ising model (5)

The problem of a flipping magnetization around a positive and negative equilibrium value can be avoided through the following tricks:

- 1 build in a restriction to e.g. positive magnetization (*Markov chain is distorted and consequences are not a priori clear*)
- 2 make averages separately for positive and negative magnetisation (*method used in IsMC.f and IsMC.py*)
- 3 make a histogram of magnetization and take peak position as “the” magnetization (*difficult in conditions with critical fluctuations ( $T \approx T_C$ )*)

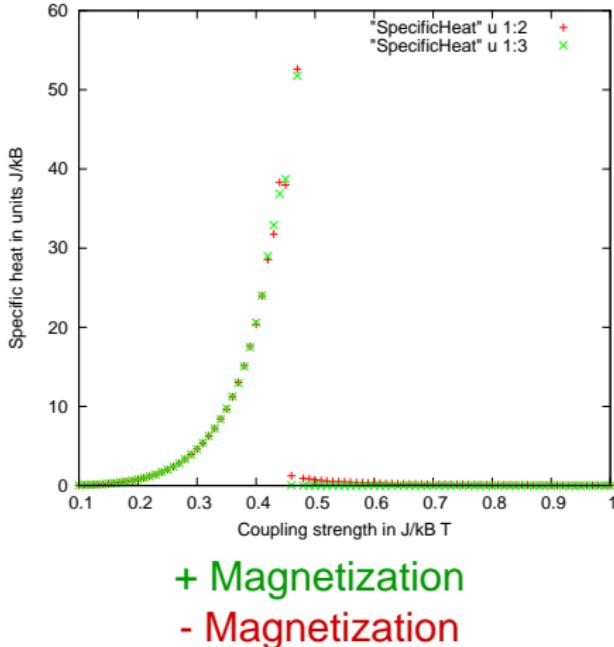
## 10.3.1 MC for the Ising model (6)

Specific heat in units  $\frac{J}{k_B}$  as a function of  $\frac{J}{k_B T}$  for  $H = 0$

- Specific heat can be computed through

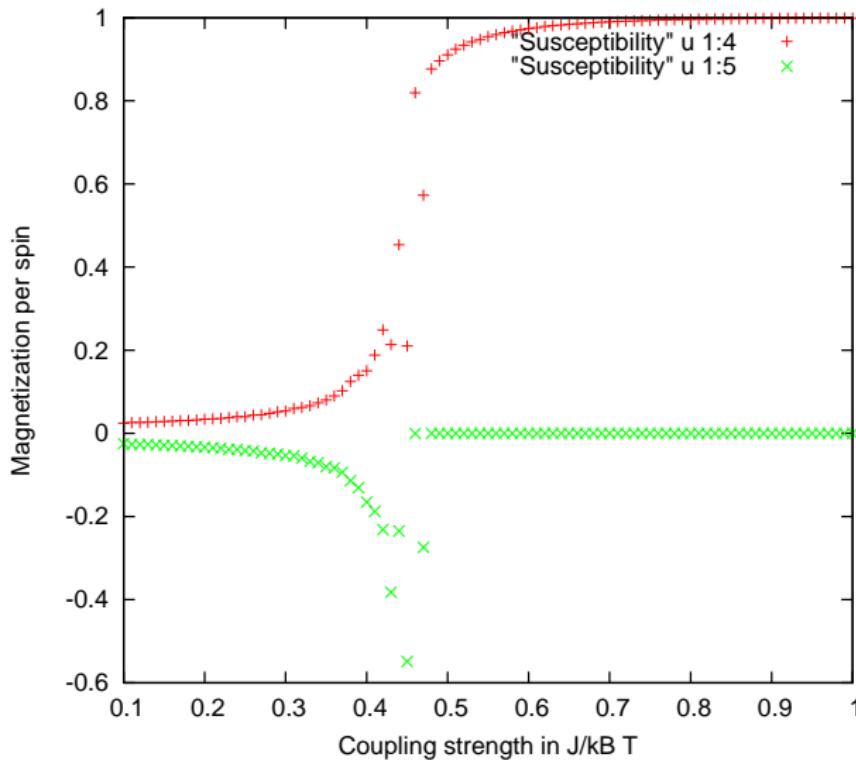
$$C_V = \frac{1}{k_B T^2} \left( \langle E^2 \rangle - \langle E \rangle^2 \right)$$

- $C_V$  exhibits a peak near  $T \approx T_c$
- “+” and “-” magnetization should provide SAME value for  $C_V$  for  $T > T_c$



## 10.3.1 MC for the Ising model (7)

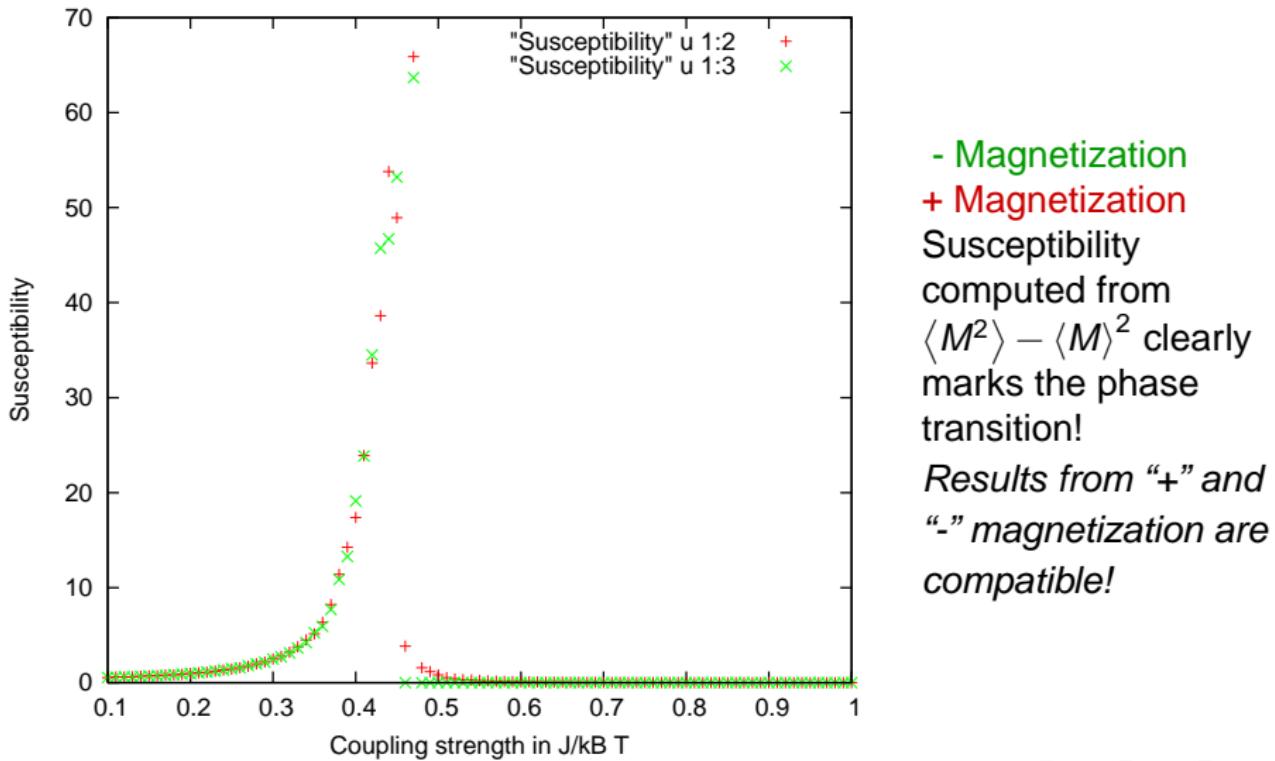
### Magnetization per spin as a function of $\frac{J}{k_B T}$ for $H = 0$



- Magnetization  
+ Magnetization  
Above the critical temperature the net magnetization is indeed zero!

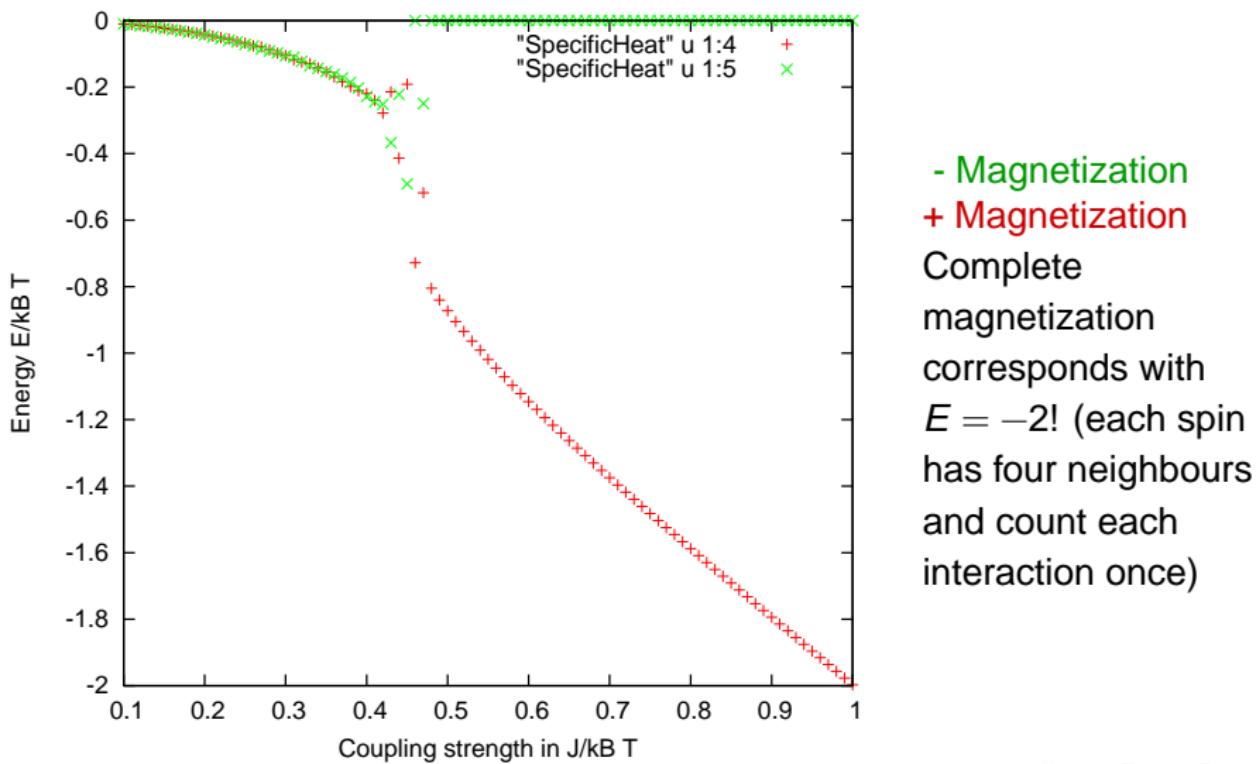
## 10.3.1 MC for the Ising model (8)

Susceptibility per spin as a function of  $\frac{J}{k_B T}$  for  $H = 0$

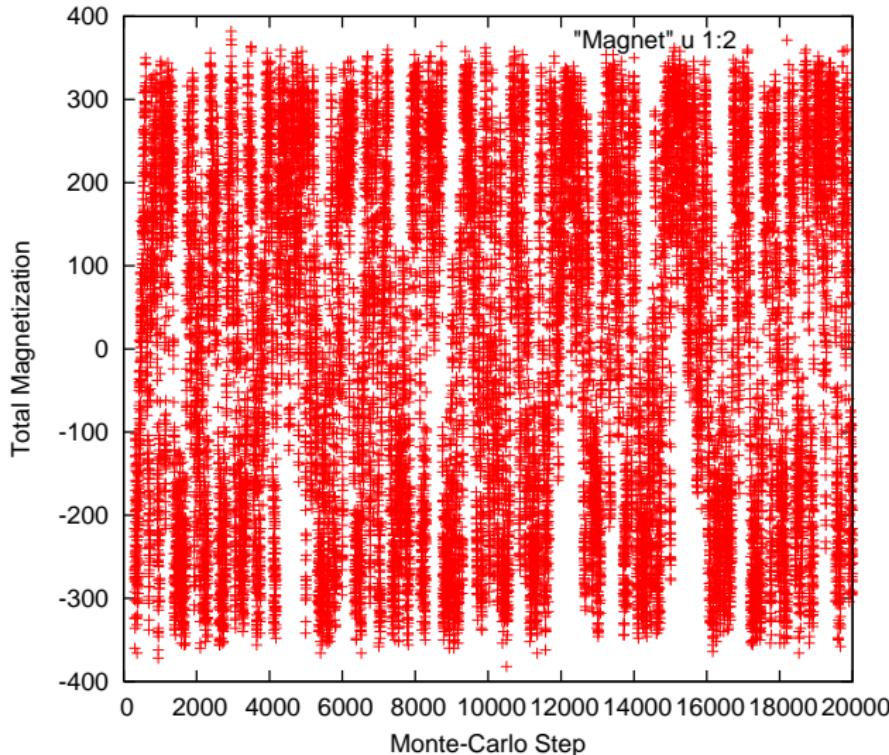


## 10.3.1 MC for the Ising model (9)

Energy per spin as a function of  $\frac{J}{k_B T}$  for  $H = 0$



## 10.3.1 MC for the Ising model (10)



Variation of the total magnetization in a MC simulation of a two-dimensional Ising system with  $20^2$  spins,  $\frac{J}{k_B T} = 0.42$ , and  $H = 0$

For  $T \approx T_C$  one observes enormous fluctuations!

### 10.3.1 MC for the Ising model (11)

**Ising model has no prescription for the dynamics and cannot be studied with MD method. Metropolis MC method makes Ising system dynamic.**

- Metropolis MC not too dissimilar from natural evolution of Ising-like system (?)
- Near the critical point one faces critical slowing down: correlation time diverges as a power of the correlation length (or, it takes the system a very long time to recover from local perturbations or fluctuations)
- Study of the dynamics near critical points allows one to study the critical behaviour but requires special techniques like finite-size scaling techniques (or, how physical observables depend on the size of the simulation system)
- Finite-size scaling based on the observation that near the critical points the correlation length approaches the size of the simulation system.

## 10.3.1 MC for the Ising model (12)

### Alternatives for the Metropolis algorithm

- Different choices for  $\omega_{XX'}$ : flip two spins which are opposite (or, total magnetization does not change)
- Heat-bath method: more efficient near  $T \approx T_c$ 
  - 1 random number generator selects a spin  $s_i$  (position of spin  $i$ )
  - 2  $s_i$  surrounded by  $n_+$  "+" spins and  $n_- = 4 - n_+$  "-" spins
  - 3 Hamiltonian for  $s_i$  in a fixed neighbour configuration  
$$\mathcal{H}(s_i | S - s_i) = -J(2n_+ - 4)s_i$$
  - 4  $s_i$  is given the "+" or "-" value with probability

$$P_{\pm} = \frac{\exp \pm (2n_+ - 4) \beta J}{\exp + (2n_+ - 4) \beta J + \exp - (2n_+ - 4) \beta J}$$

The heat-bath method has a smaller correlation time than the Metropolis method! (more efficient near the phase transition)

## 10.3.2 MC simulation for a monoatomic gas (1)

- Metropolis MC: enables to compute averages of static quantities
- In classical systems: momentum d.o.f. can be integrated out

$$\frac{V}{h^3} \int d\vec{p} \exp -\frac{\beta p^2}{2m} = V \left( \frac{2\pi m k_B T}{h^2} \right)^{\frac{3}{2}} = \frac{V}{\Lambda^3}$$

$\Lambda$  is the thermal wavelength

- For dynamical variables  $A(R)$ : Boltzmann factor depends only on configurational potential energy

$$\rho(R) = \exp [-\beta U(R)] \quad R \equiv (\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$$

$U(R) = \sum_{i < j} U(|\vec{r}_i - \vec{r}_j|)$  is the total interaction energy of the system

## 10.3.2 MC simulation for a monoatomic gas (2)

MC algorithm for a monoatomic gas:  $T(X \rightarrow X') = \omega_{XX'} A_{X'X}$

■ ?? $\omega_{XX'}$ ??

- select a particle at random
- move it within a cube centered at the old position of the particle (*the position of the remaining particles is kept fixed*)

■ ?? $A_{XX'}$ ??

- if  $E(X') < E(X)$  trial configuration is accepted
- if  $E(X') > E(X)$  trial configuration is accepted with probability

$$0 < A_{XX'} = \exp -\beta \Delta E (X \rightarrow X') = \exp -\beta (E(X') - E(X)) < 1$$

The algorithm obeys the requirements of a Markov chain:

1 detailed balance ( $T(X \rightarrow X') = \omega_{XX'} A_{X'X}; \omega_{X'X} = \omega_{XX'}$ )

$$\frac{A_{XX'}}{A_{X'X}} = \frac{\rho(X')}{\rho(X)} = \frac{\exp -\beta E(X')}{\exp -\beta E(X)}$$

2 aperiodicity

3 connectedness

## 10.3.2 MC simulation for a monoatomic gas (3)

### Good choice for the displacement volume??

- any cube fulfills aperiodicity, connectedness and “detailed balance” conditions
- efficiency grows as one can generate more and more statistically independent solutions in a certain amount of simulation time
  - cube too small: many MC steps before arriving at statistically independent configuration
  - cube too large: particles make large moves and energy will increase in majority of cases
  - rule of thumb: choose the cube such that  $0.4 \leq A_{xx'} \leq 0.6$
- for hard spheres good efficiency is difficult to reach
  - very strong repulsion if particles move too close (large  $\Delta E$ )
  - optimized acceptance rate is low  $A_{xx'} \approx 0.1$

## 10.3.2 MC simulation for a monoatomic gas (4)

### Computer code for Metropolis method for argon

- much of the MD code can be copied (e.g. *particles are released from fcc lattice positions*)
- trial displacement should respect PBC!
- trial displacement in the  $x$  direction (without correcting for PBC)

$$x_{new} - x_{old} = d_{max} (2r - 1) \quad d_{max} = \frac{L}{2}$$

$r$  is a random number in  $[0,1]$

- potential energy is updated after every acceptance of a trial configuration
- one needs a list of neighbours within a range  $r_{cut-off}$  for each particle
- only quantities depending on  $(\vec{r}_i, i = 1, \dots, N)$  can be computed (the momenta  $(\vec{p}_i, i = 1, \dots, N)$ ) CANNOT be considered in the MC method)

## 10.3.2 MC simulation for a monoatomic gas (5)

### Computer code for Metropolis method for argon: Initialise the calculation

- 1 Specify number of particles (PartNum) in the box ( $4N^3$ )
- 2 User input

```
2.5      # Temperature in reduced units
300      # No. of MCS equilibration steps (InitStep)
10       # MCS between rescaling displacement volume
10       # MCS between Update of the pair list
3500     # No. of MCS production steps (SimStep)
1.05     # Density in reduced units
448957   # Seed of the random generator
10       # MCS between output of data
```

- 3 Initialize the positions of the particles (FCC lattice)
- 4 Run the Metropolis Monte-Carlo procedure for InitStep steps (1 MCS is one sweep over ALL particles)

## 10.3.2 MC simulation for a monoatomic gas (6)

### Computer code for Metropolis method for argon: Metropolis Monte-Carlo

#### LOOP OVER THE REQUIRED AMOUNT OF MCS

- 1 for each MCS: sweep over all particles  
`CALL MCSweep(AcceptRate)`
- 2 after each MCS: store whereabouts of particle “27”
- 3 if in production phase: write the results of the calculations  
`CALL OutputMC(Step, EquilPhase)`
- 4 if in equilibration phase: every 10 MCS or so, scale the displacement volume so as to make the acceptance rate about 10 - 50 %  
`CALL ScaleDisplVol(AcceptRate)`
- 5 every 10 MCS or so: update the pair list of interacting molecules `CALL UpdatePairList`

#### END OF LOOP OVER THE REQUIRED AMOUNT OF MCS

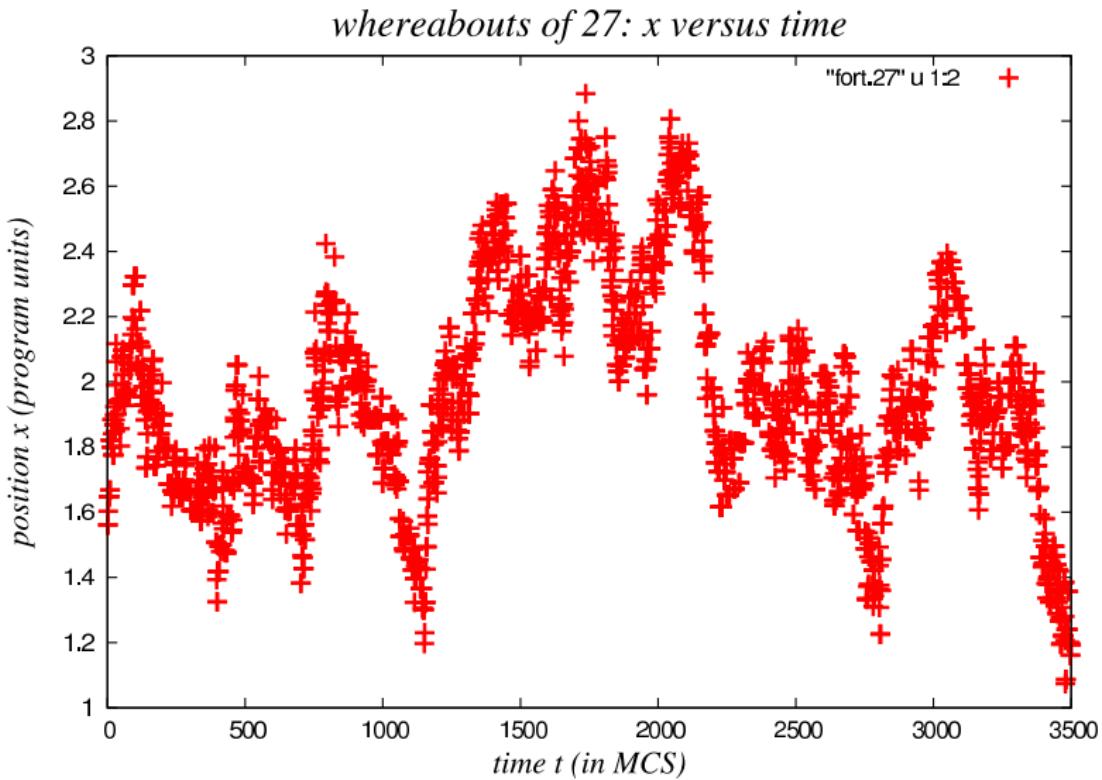
## 10.3.2 MC simulation for a monoatomic gas (6)

### Output of the Computer code for Metropolis method for argon: Metropolis Monte-Carlo

```
file = 'potential', MCStep (production phase), < E_{pot} >
file = 'virial' , MCStep (production phase), virial
file = 'correl' , r (program units), g(r)
file = 'whereabout27', MCStep (production phase), x_{27},
                  y_{27}, z_{27}
file = 'initialpositions', particle number, x, y, z
file = 'r2versustime', MCStep (production phase), <r^2>
```

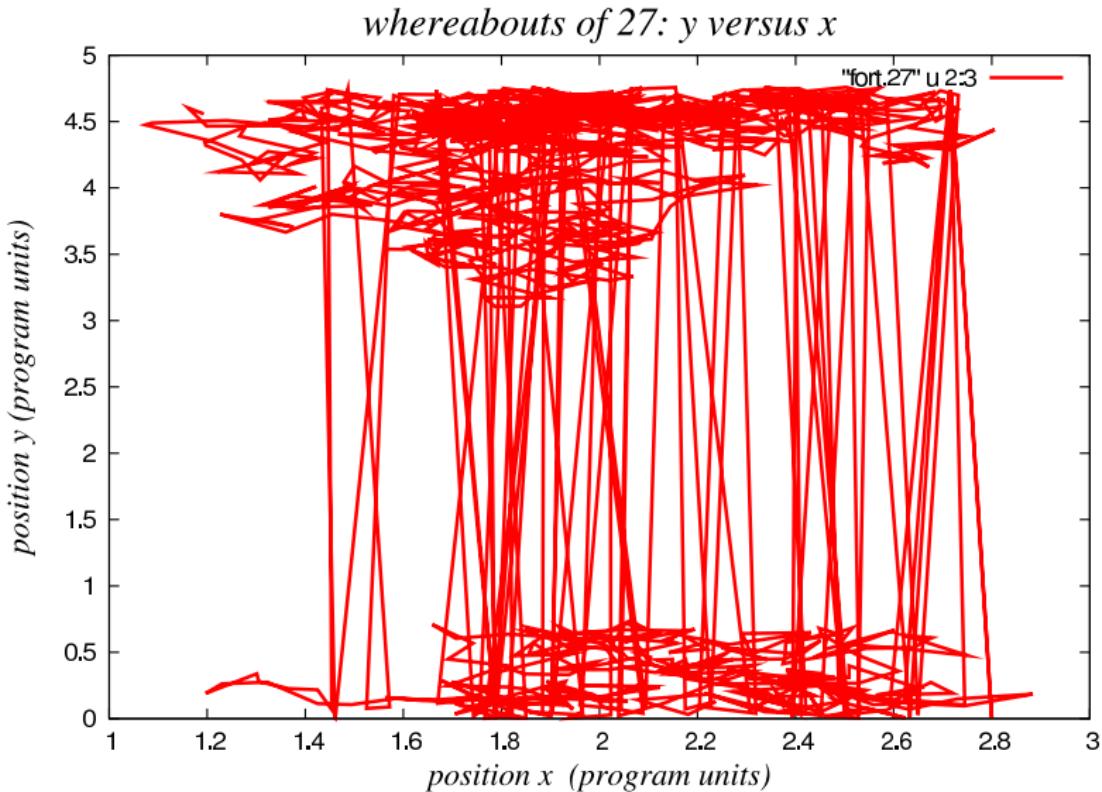
## 10.3.2 MC simulation for a monoatomic gas (7)

WHEREABOUTS of particle number 27 ( $T=1, \rho=1$ )



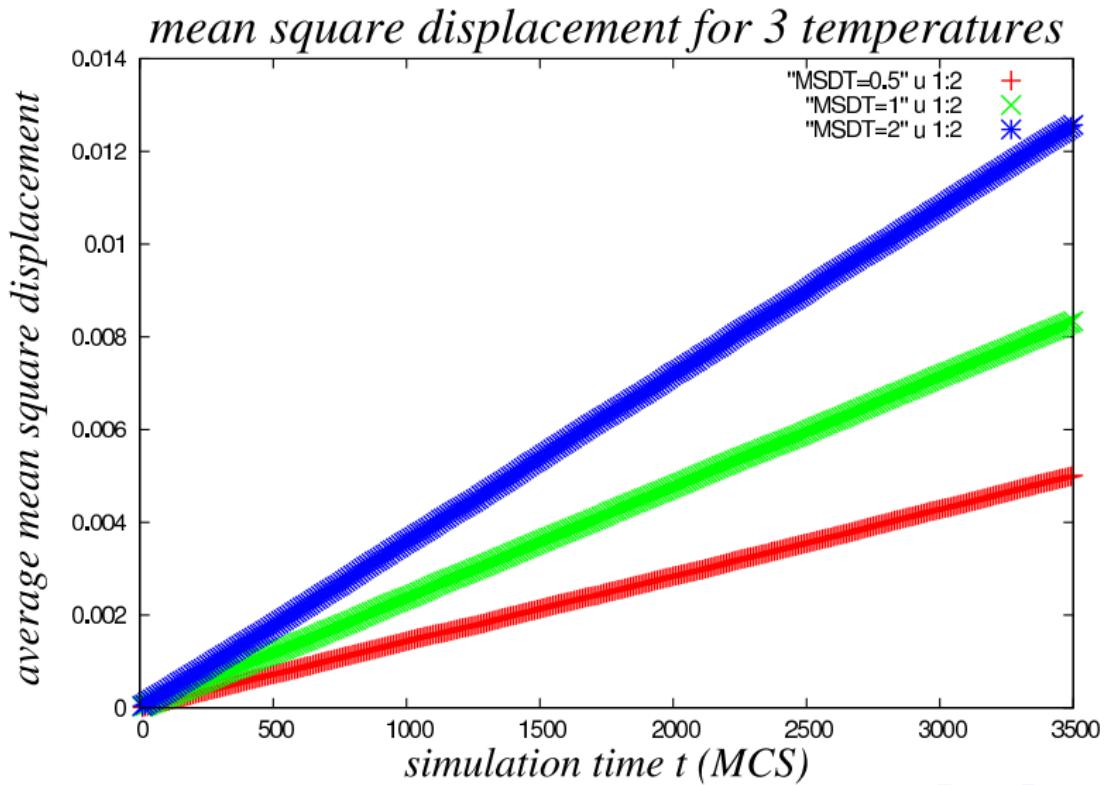
## 10.3.2 MC simulation for a monoatomic gas (7)

WHEREABOUTS of particle number 27 ( $T=1, \rho=1$ )



## 10.3.2 MC simulation for a monoatomic gas (8)

Diffusion properties (mean-square displacement per particle versus MCS - time) for  $T = 0.5$ ,  $T = 1$ ,  $T = 2$  and  $\rho = 1$



## 10.4 Other ensembles

- The canonical ensemble  $\rho(X) = \exp -\beta \mathcal{H}(X)$  is the most natural ensemble for MC simulations.
- There exists a microcanonical MC method: seldomly used as it is of little practical importance
- It is possible to use the Metropolis Monte-Carlo to simulate other ensembles

10.4.1 The *NPT* ensemble (isothermal-isobaric ensemble)

10.4.2 The grand canonical ensemble

10.4.3 The Gibbs ensemble

## 10.4.1 The (*NPT*) ensemble (p310) (1)

- in the (*NPT*) ensemble the ensemble average of a physical quantity  $A(R) = A(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$  is given by

$$\langle A \rangle_{(NPT)} = \frac{\frac{1}{N!} \frac{1}{\Lambda^{3N}} \int_0^\infty dV \exp -\beta PV \int dR A(R) \exp -\beta U(R)}{Q(N, P, T)}$$

Gibbs free energy:  $G(T, P, N) = -k_B T \ln Q(N, P, T)$

$$G(T, P, N) = -k_B T \ln \frac{1}{N!} \frac{1}{\Lambda^{3N}} \int_0^\infty dV \exp -\beta PV \int dR \exp -\beta U(R)$$

- evaluation of  $\langle A \rangle_{(NPT)}$  requires a Markov chain in which particle moves ( $dR$ ) AND volume changes ( $dV$ ) are allowed
- change in the volume requires appropriate rescaling of the particle positions

## 10.4.1 The (*NPT*) ensemble (2)

### Rescaling of the particle positions

- consider a cubic volume  $L \times L \times L$  volume
- rescale particle positions  $\vec{r}_i$  such that the  $\vec{s}_i$  lie in the unit volume

$$\vec{r}_i = L \vec{s}_i \quad i = 1, 2, \dots, N$$

- an ensemble average can then be rewritten as ( $d\vec{r}_i = L^3 d\vec{s}_i$ )

$$\langle A \rangle_{(NPT)} = \frac{\frac{1}{N!} \frac{1}{\Lambda^{3N}} \int_0^\infty dV \exp -\beta PV \int dS V^N A(LS) \exp -\beta U(LS)}{Q(N, P, T)}$$

with,  $S \equiv (\vec{s}_1, \vec{s}_2, \dots, \vec{s}_N)$  (positions within the unit cube)

- in the (*NPT*) ensemble the Boltzmann weight is given by

$$\rho(X) = \rho(V, S) = \exp -\beta PV \times V^N \times \exp -\beta U(LS)$$

## 10.4.1 The (*NPT*) ensemble (3)

### Rescaling of the particle positions (continued)

- step in the Metropolis Markov chain: either a particle move or a volume change (*determines the trial step probability*  $\omega_{XX'}$ )
- *the acceptance probability*  $A_{XX'}$  for a particle move  $S_{old} \rightarrow S_{new}$

$$\frac{\rho(X')}{\rho(X)} = \frac{\rho(V, S_{new})}{\rho(V, S_{old})} = \exp -\beta [U(LS_{new}) - U(LS_{old})]$$

- *the acceptance probability*  $A_{XX'}$  for a volume change  $V_{old} \rightarrow V_{new}$

$$\begin{aligned} \frac{\rho(X')}{\rho(X)} &= \frac{\rho(V_{new}, S)}{\rho(V_{old}, S)} = \exp -\beta [U(L_{new}S) - U(L_{old}S)] \\ &\times \left( \frac{V_{new}}{V_{old}} \right)^N \exp -\beta [PV_{new} - PV_{old}] \end{aligned}$$

## 10.4.1 The (*NPT*) ensemble (4)

### Rescaling of the particle positions (continued)

- calculation of the potential energy difference associated with a volume change can be computationally very demanding (AS IT MAY REQUIRE A SWEEP OVER ALL PAIRS OF MOLECULES)
- potential-energy difference for  $L_{\text{before}} \rightarrow L_{\text{after}}$ : relatively easy for potentials of the type  $U(r) = \left(\frac{\sigma}{r}\right)^k$

$$\frac{\sigma^k}{\sum_{i < j} (L_{\text{after}} s_{ij})^k} = \left( \frac{L_{\text{before}}}{L_{\text{after}}} \right)^k \frac{\sigma^k}{\sum_{i < j} (L_{\text{before}} s_{ij})^k}$$

### OVERALL SCALING FACTOR!!

- if method cannot be applied: volume changes are expensive and are attempted at a lower rate than particle moves

## 10.4.1 The (*NPT*) ensemble (5)

### Rescaling of the particle positions (continued)

- when computing the average interaction energy corrections are needed if potential is cut off beyond  $r_{cut-off}$

$$\langle U \rangle = \langle U \rangle_{r < cut-off} + \frac{2\pi N(N-1)}{V} \int_{r_{cut-off}}^{\infty} dr r^2 g(r) U(r)$$

- usually one puts  $g(r) \approx 1$  in the second term: second term does not contribute to the total energy difference in MC steps
- for a potential consisting of powers:  $r_{cut-off} \equiv L s_{cut-off}$

## 10.4.2 The grand canonical ensemble (1)

The temperature  $T$ , the system volume  $V$  and the chemical potential  $\mu$  are given, the particle number AND pressure vary

- average of a configurational physical quantity  $A(R)$

$$\langle A(R) \rangle_{\mu VT} = \frac{\sum_{N=0}^{+\infty} \frac{1}{N!} \exp \beta \mu N \frac{1}{\Lambda^{3N}} \int dR_N A(R_N) \exp -\beta U(R_N)}{Z_G(\mu, V, T)}$$

$\Lambda = \sqrt{\frac{\hbar^2}{2\pi m k_B T}}$  the THERMAL WAVELENGTH

$R_N = R(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$ : the positions of the  $N$  particles

- the  $\Lambda^{3N}$  appears from integrating out the momentum d.o.f.
- Metropolis algorithm:
  - 1 random particle **moves**
  - 2 particle **creations** at random positions
  - 3 **annihilations** of randomly chosen particles

## 10.4.2 The grand canonical ensemble (2)

### Metropolis MC algorithm in the grand canonical ensemble

- $\omega_{XX'}$ ?
  - creation, annihilation or move?  
divide  $[0, 1]$  in three segments with sizes equal to the respective probabilities
  - segment for annihilation and creation must be equal to keep  $\omega_{XX'}$  symmetric
  - generate random number and check the segment in which this number lies
- $A_{XX'}$ ?  
Weight factor replacing the Boltzmann one

$$\rho(X) \equiv \rho(N, R_N) = \frac{1}{\Lambda^{3N} N!} \exp -\beta U(R_N) \exp \beta \mu N$$

- if  $\rho(X') > \rho(X)$  trial configuration is accepted
- if  $\rho(X') < \rho(X)$  trial configuration is accepted with probability

$$A_{XX'} = \frac{\rho(X')}{\rho(X)}$$

## 10.4.2 The grand canonical ensemble (3)

- **if a creation is attempted** ( $NR_N \rightarrow N+1R_{N+1}$ )
  - select a random position and compute the “extra” potential energy of a particle at that position with the remaining ones
  - the creation of a particle is accepted with probability

$$A_{XX'} = \frac{\rho(N+1, R_{N+1})}{\rho(N, R_N)} = \frac{V}{\Lambda^3} \frac{\exp \beta \mu}{N+1} \exp -\beta [U(R_{N+1}) - U(R_N)]$$

$V$ : accounts for the probability that a particle is created at  $d^3 r_{N+1}$ ;  
this probability is given by  $\frac{d^3 r_{N+1}}{V}$  ( $Z_G = \sum_N \int dR_N \rho(N, R_N)$ )

- **if an annihilation is attempted** ( $NR_N \rightarrow N-1R_{N-1}$ )
  - the annihilation of a particle is accepted with probability

$$A_{XX'} = \frac{\rho(N-1, R_{N-1})}{\rho(N, R_N)} = \frac{\Lambda^3}{V} N \exp -\beta \mu \exp -\beta [U(R_{N-1}) - U(R_N)]$$

- **attempted particle moves** are processed similarly to the canonical case (section 10.3.2)

## 10.4.2 The grand canonical ensemble (4)

- other approaches: run the simulation with “ghost” particles which can be switched on and off (“vacuum fluctuations”)
- the chemical potential is a negative quantity  $\mu = -|\mu|$  for bosons and fermions in the classical regime
- if the chemical potential is large (and negative!): small chance of creating extra particles due to the presence of the  $\exp \beta \mu$  in  $A_{XX'}$ 
  - *change the relative rates of creation and annihilation and correct for this by a suitable change in the  $A_{XX'}$*
  - *remember that detailed balance requires that*  
 $T(X \rightarrow X') = T(X' \rightarrow X)$
- at high density regions insertion of new particles is likely to fail due to the  $\exp -\beta [U(R_{N+1}) - U(R_N)]$  term
  - *methods have been devised for locating cavities in fluids and creating particles preferentially in these regions*
  - particles are likely to flow from “high” to “low” density regions

## 10.4.3 The Gibbs ensemble (1)

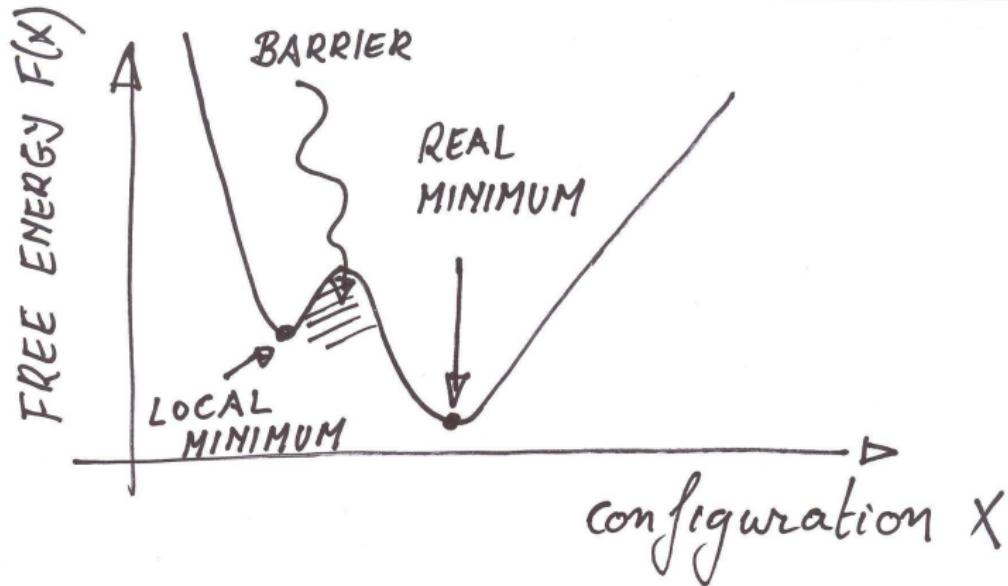
- Gibbs ensemble: MC method that is well suited for
  - 1 studying the coexistence of different phases of the same material
  - 2 different species that can transform into each other (chemical or nuclear reactions)
- for fixed temperature  $T$  and pressure  $P$ 
  - initial conditions: chemical potentials will not be equal (which corresponds with non-equilibrium conditions!)
  - one species will grow at the expense of the other
  - equilibrium: equal chemical potentials or complete disappearance of one species
  - in practice: equilibrium difficult to achieve due to free-energy barriers

## 10.4.3 The Gibbs ensemble (2)

- in simulation system: two subsystems that are free to exchange particles (avoids free-energy barriers)
- total system ( $N, V$ : fixed) is subdivided into two subsystems
  - 1 variable volumes:  $V_1 + V_2 = V$
  - 2 variable amount of particles:  $N_1 + N_2 = N$
- total system is ( $N, V, T$ ) ensemble
- no interactions between particles “1” and particles “2”
- interactions between the particles of the same species

### 10.4.3 The Gibbs ensemble (3)

Very often: simulations will get stuck in local minima of the free energy (most often in situations of phase coexistence, e.g. undercooled water)



Both nature and simulations experience difficulties in overcoming free-energy barriers!

## 10.4.3 The Gibbs ensemble (4)

### Metropolis MC algorithm for the Gibbs ensemble

- $\omega_{XX'}$ 
  - trial state
    - 1 move in subsystem 1 or move in subsystem 2
    - 2 particle transfer from 1 to 2 or from 2 to 1
    - 3 volume change in 1 or volume change in 2
  - $A_{XX'}$  in terms of the Weight factor

$$\begin{aligned}\rho(X) &\equiv \rho(V_1, N_1, S_1, V_2, N_2, S_2) \\ &= \frac{V_1^{N_1}}{N_1!} \frac{V_2^{N_2}}{N_2!} \exp -\beta U(L_1 S_1) \exp -\beta U(L_2 S_2)\end{aligned}$$

- if  $\rho(X') > \rho(X)$  trial configuration is accepted
- if  $\rho(X') < \rho(X)$  trial configuration is accepted with probability

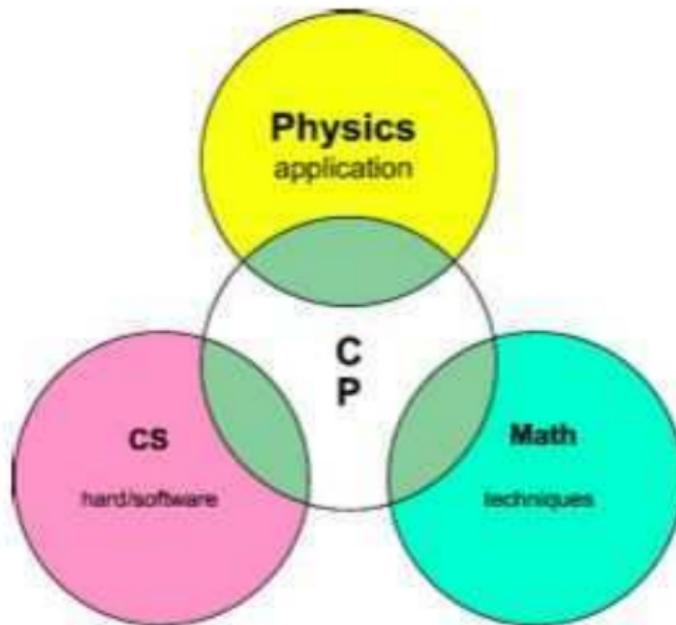
$$A_{XX'} = \frac{\rho(X')}{\rho(X)}$$

# Computational physics: epilogue ..

COMPUTATIONAL PHYSICS (CP): UNION

OF PHYSICS WITH COMPUTER SCIENCE

(CS) AND APPLIED MATHEMATICS

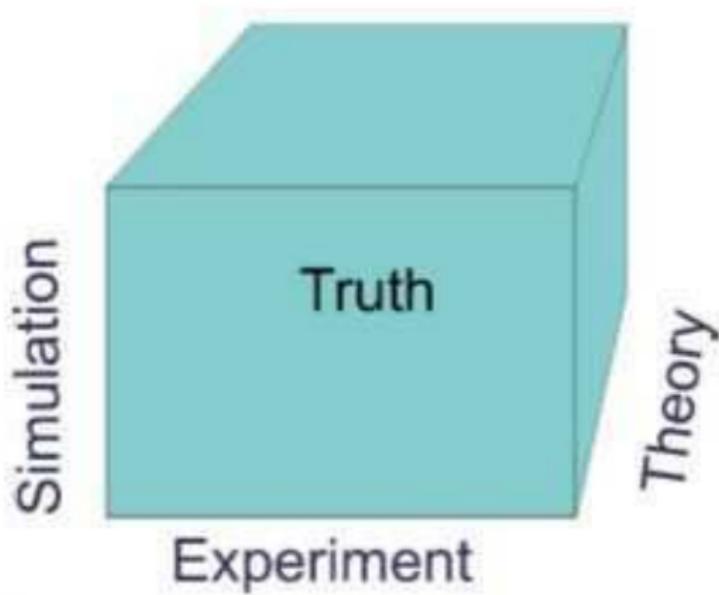


# Computational physics: epilogue ..

COMPUTATIONAL PHYSICS: ADDITION OF

SIMULATION AS A PATH TO SCIENTIFIC

TRUTH

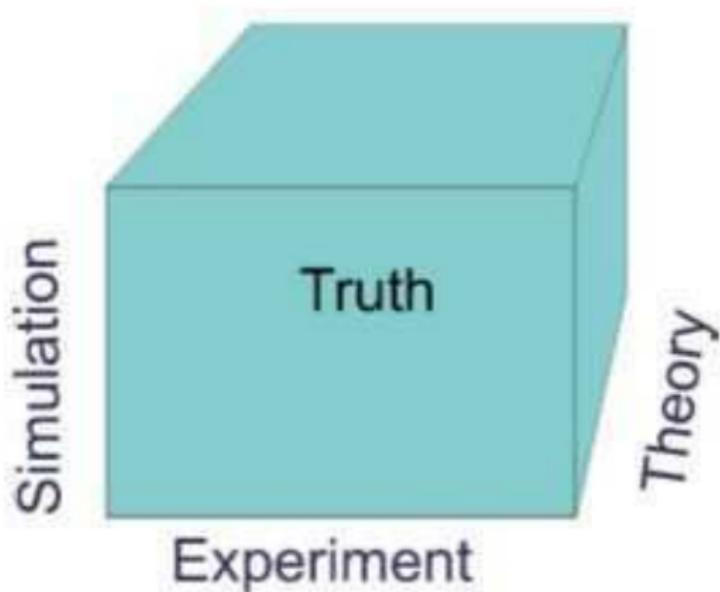


# Computational physics: epilogue ..

COMPUTATIONAL PHYSICS: ADDITION OF

SIMULATION AS A PATH TO SCIENTIFIC

TRUTH



# Computational Physics: Density Functional Theory (DFT) Part 1

Toon Verstraelen

[toon.verstraelen@ugent.be](mailto:toon.verstraelen@ugent.be)

Ghent University



Academic Year 2012-2013, First Semester

## Outline

- 1 Introduction
- 2 Hartree-Fock Method
- 3 Density (matrix)
- 4 HF Limitations
- 5 Density Functional Theory

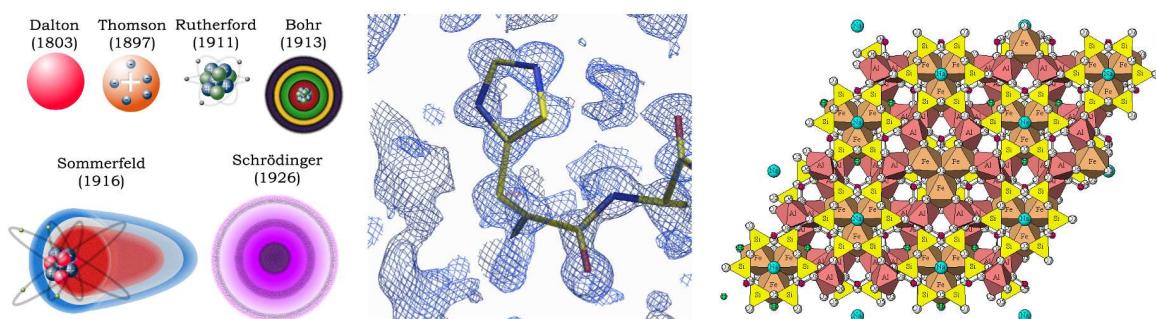
# What is Density Functional Theory (DFT)

A Computationally **efficient** method  
to solve **approximately**  
the **quantummechanical electronic many-body** problem

- Single electron: straightforward variational solution of the Schrödinger equation
- Many electrons: conventional PDE solvers not feasible
  - $e^-$  are fermions, hence antisymmetric wavefunction (Pauli repulsion, exchange effect)
  - $e^-$  exert mutual repulsion (correlation effect)
- In computational applications
  - Many hundreds of electrons
  - External potential is due to atomic nuclei

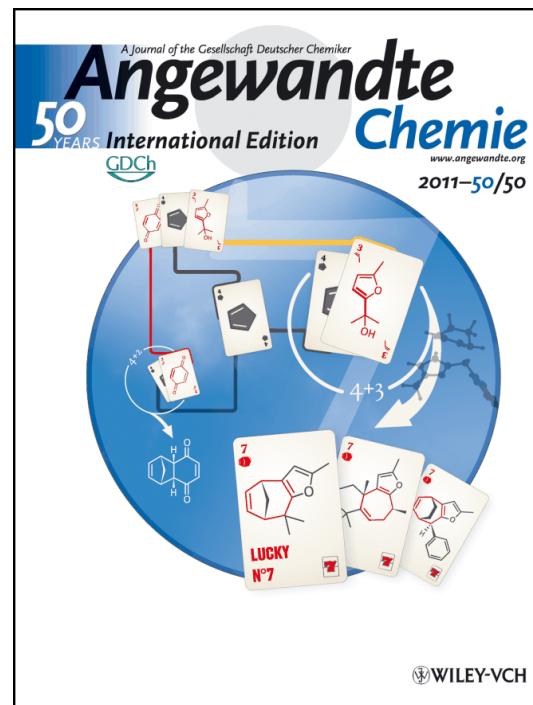
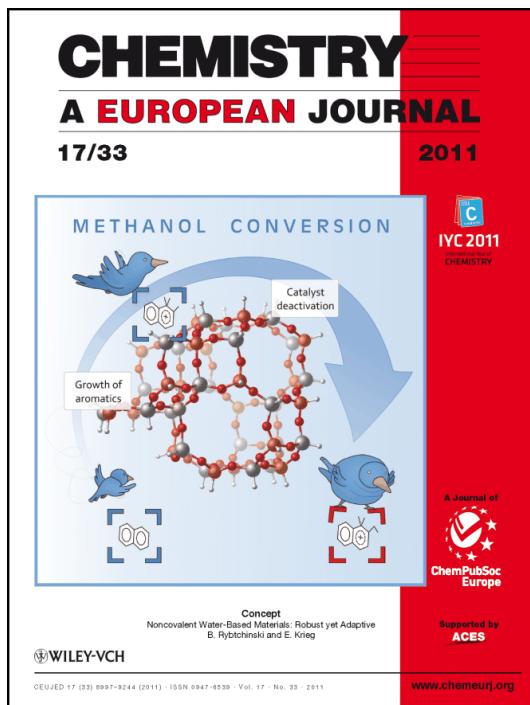
3/99

# Why is DFT important? (1)



- Electronic structure of atoms, molecules and periodic systems
- Potential energy surface for MD simulations
- Computational physics approach to chemistry

## Why is DFT important? (2)



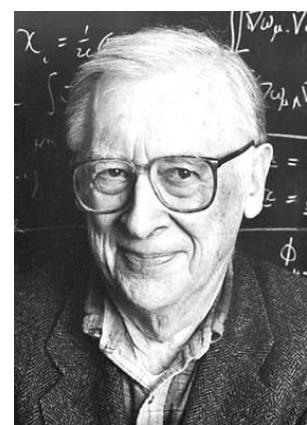
5/99

## Why is DFT important? (3)

The 1998 Nobel prize in Chemistry was given to a theoretical physicist and a theoretical chemist/mathematician!

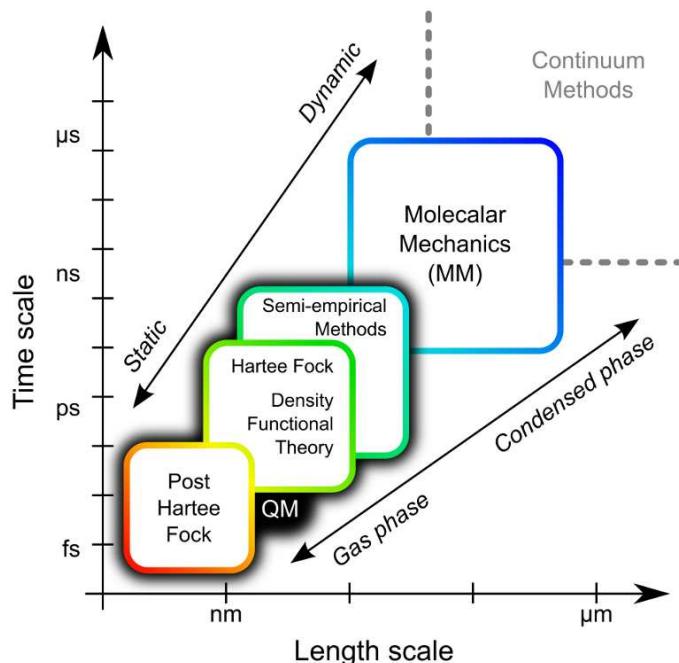


Walter Kohn "for his development of the density-functional theory"



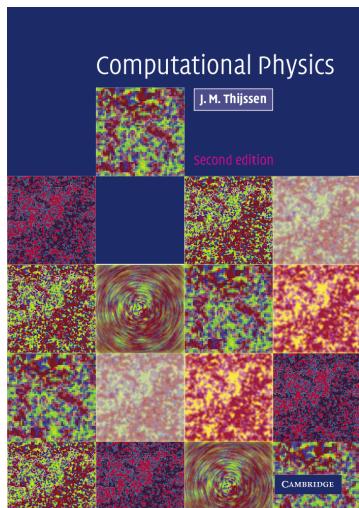
John A. Pople "for his development of computational methods in quantum chemistry"

# The grand scheme of molecular potentials



7/99

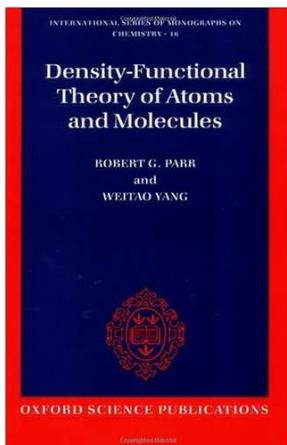
## Parts of the book



J.M. Thijssen  
 "Computational Physics",  
 second edition (2007)

- Chapter 1 : Introduction (p1-13)
- Chapter 2 : Quantum scattering with a spherically symmetric potential
- Chapter 3 : The variational method for the Schrödinger equation
- Chapter 4 : The Hartree-Fock method
- Chapter 5 : Density functional theory
- ...

# ... and another book



R.G. Parr & W. Yang  
 "Density-Functional  
 Theory of Atoms and  
 Molecules", (1989)

- No compulsory material.
- Provides more theoretical background than Thijssen.
- Widely recognized as the DFT bible.
- All theory needed for the exam is discussed in the slides.
- Notation in slides is compatible with Thijssen.

9/99

# Conventions and approach

- Atomic units are used in theory and implementation ( $m_e = 1$ ,  $e = 1$ ,  $\hbar = 1$ ,  $\frac{1}{4\pi\epsilon_0} = 1$ )
- This is a computational physics course, i.e. theory is kept minimal.
- $\int d\mathbf{x}_i = \sum_{s_i} \int d\mathbf{r}_i$
- Compact notation
  - $\sum_{i=1}^N = \sum_i^N$
  - $\sum_{i=1}^N \sum_{j=1}^N = \sum_{ij}^N$
  - $\sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N = \sum_{i \neq j}^N$

# Functionals and functional derivatives (1)

A functional is formally *a function of a function*. For example, the total number of electrons is a functional of the density:

$$N[n(\mathbf{r})] = \int n(\mathbf{r}) d\mathbf{r}$$

the derivative of a functional is a generalization of the gradient of a multivariate function. It is defined implicitly:

$$\int \frac{\delta F[f]}{\delta f(\mathbf{r})} g(\mathbf{r}) d\mathbf{r} = \left. \frac{dF[f(\mathbf{r}) + \epsilon g(\mathbf{r})]}{d\epsilon} \right|_{\epsilon=0}$$

Functional derivatives are typically used to define a stationary point of a functional, e.g.

$$\frac{\delta F[f]}{\delta f(\mathbf{r})} = 0$$

11/99

# Functionals and functional derivatives (2)

When the functional  $F$  has the form

$$F[n(\mathbf{r}), \nabla n(\mathbf{r})] = \int f(n(\mathbf{r}), \nabla n(\mathbf{r})) d\mathbf{r}$$

The functional derivative of  $F$  towards  $n(\mathbf{r})$  is formally given by:

$$\frac{\delta F[n(\mathbf{r}), \nabla n(\mathbf{r})]}{\delta n(\mathbf{r})} = \frac{\partial f}{\partial n}(\mathbf{r}) - \nabla \cdot \left( \frac{\partial f}{\partial \nabla n}(\mathbf{r}) \right)$$

# Optimizing a real function of a complex variable

This problem is discussed in detail in the auxiliary notes on Minerva!

When a real function,  $f$ , depends on a complex variable,  $z$ , e.g. the energy as function of a wavefunction, a stationary point is characterized by:

$$\frac{\partial f}{\partial z_r} = 0 \quad \text{and} \quad \frac{\partial f}{\partial z_i} = 0$$

where  $z = z_r + iz_i$ . One can show (see auxiliary notes) that the following condition is equivalent when the partial derivatives are taken such that  $z$  and its complex conjugate,  $z^*$ , are treated as independent variables:

$$\frac{\partial f}{\partial z^*} = 0 \quad \text{or} \quad \frac{\partial f}{\partial z} = 0$$

13/99

# The non-relativistic molecular Hamiltonian

$$H = -\frac{1}{2} \sum_i^N \nabla_i^2 - \sum_i^N \sum_A^M \frac{Z_A}{r_{iA}} + \frac{1}{2} \sum_{i \neq j}^N \frac{1}{r_{ij}} + \frac{1}{2} \sum_{A \neq B}^M \frac{Z_A Z_B}{R_{AB}} - \sum_A^M \frac{\nabla_A^2}{2m_A}$$

- number of electrons:  $N$
- number of nuclei:  $M$
- electron indexes:  $i, j$
- nuclear indexes:  $A, B$
- nuclear charges:  $Z_A, Z_B$
- nuclear mass:  $m_A, m_B$
- inter particle distances:  $r_{iA}, r_{ij}, R_{AB}$

14/99

# Separation of nuclear and electronic Hamiltonians

## Assumptions

- Electrons are much lighter (faster) than nuclei
- As nuclei move,  $e^-$  instantaneously adopt ground state
- Treat nuclear coordinates as parameters in the electronic Hamiltonian.  $H_e = -\frac{1}{2} \sum_i^N \nabla_i^2 - \sum_i^N \sum_A^M \frac{Z_A}{r_{iA}} + \frac{1}{2} \sum_{i \neq j}^N \frac{1}{r_{ij}}$
- Compute the electronic ground state energy  $E_{gs,e}$ .
- Obtain potential felt by nuclei.  $V_n = E_{gs,e} + \frac{1}{2} \sum_{A \neq B}^M \frac{Z_A Z_B}{R_{AB}}$
- Approximate nuclei as (classical) point particles in a potential governed by the electronic structure.  

$$H_n = \sum_A^M \frac{P_A^2}{2m_A} + V_n(\mathbf{R}_1, \dots, \mathbf{R}_M)$$

15/99

# The electronic Hamiltonian

All electronic structure methods approximate ground and excited states of the quantum mechanical electronic many-body Hamiltonian:

$$H_e = -\frac{1}{2} \sum_i^N \nabla_i^2 - \sum_i^N \sum_A^M \frac{Z_A}{r_{iA}} + \frac{1}{2} \sum_{i \neq j}^N \frac{1}{r_{ij}}$$

# The electronic wavefunction

The Pauli principle for fermions implies an antisymmetric wavefunction, i.e.

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = -\Psi(\mathbf{x}_2, \mathbf{x}_1, \dots, \mathbf{x}_N).$$

where  $\mathbf{x}_i$  contains both Cartesian and spin coordinates.

If one would allow

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \Psi(\mathbf{x}_2, \mathbf{x}_1, \dots, \mathbf{x}_N),$$

then particle 1 and 2 would occupy the same state, which is forbidden.

17/99

# The Slater determinant

The Slater determinant is a mathematical construct:

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) & \cdots & \psi_N(\mathbf{x}_1) \\ \psi_1(\mathbf{x}_2) & \psi_2(\mathbf{x}_2) & \cdots & \psi_N(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ \psi_1(\mathbf{x}_N) & \psi_2(\mathbf{x}_N) & \cdots & \psi_N(\mathbf{x}_N) \end{vmatrix}$$

- If  $\psi_i == \psi_j$ , the determinant disappears.
- If  $\psi_i$  and  $\psi_j$  are swapped, the determinant changes sign.
- $\psi_i$  are orthonormal one-body wavefunctions, a.k.a orbitals.
- A unitary transformation of the orbitals only affects the phase of the Slater determinant.
- Not all N-fermion wavefunctions can be written as Slater determinants. For example, linear combinations of Slater determinants also satisfy the antisymmetry condition.

18/99

# Slater-Condon rules (1)

Mathematical expressions for expectation values of one- and two-body operators for a Slater determinant.

- QM operators generally only involve one or two particles, not all N particles at the same time.
- one-body operators
  - The probability density of electron  $i$  at point  $\mathbf{r}$ :  
 $\hat{n}(i; \mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}_i)$
  - Kinetic energy of electron  $i$ :  $\hat{T}(i) = -\frac{1}{2}\nabla_i^2$
  - Interaction of electron  $i$  with the nuclei:  
 $\hat{V}_{\text{ext}}(i) = V_{\text{ext}}(\mathbf{r}_i) = -\sum_A^M \frac{1}{|\mathbf{r}_i - \mathbf{r}_A|}$
- two-body operators
  - Coulomb repulsion between electron  $i$  and  $j$ :  $\hat{V}_{\text{ee}}(ij) = \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$

19/99

# Slater-Condon rules (2)

One-body operator for N particles:

$$\hat{A} = \sum_{i=1}^N \hat{a}(i)$$

The expectation value becomes:

$$\begin{aligned} \langle \Psi | \hat{A} | \Psi \rangle &= \sum_i^N \langle \Psi | \hat{a}(i) | \Psi \rangle \\ &= \sum_i^N \langle \psi_i | \hat{a}(1) | \psi_i \rangle \\ &= \sum_i^N \int \psi_i^*(\mathbf{x}_1) \hat{a}(\mathbf{x}_1) \psi_i(\mathbf{x}_1) d\mathbf{x}_1 \end{aligned}$$

20/99

# Slater-Condon rules (3)

Two-body operator for N particles:

$$\hat{B} = \frac{1}{2} \sum_{i \neq j}^N \hat{b}(i, j)$$

The expectation value becomes:

$$\begin{aligned} \langle \Psi | \hat{O} | \Psi \rangle &= \frac{1}{2} \sum_{ij}^N \langle \psi_i \psi_j | \hat{b}(1, 2) | \psi_i \psi_j \rangle - \langle \psi_i \psi_j | \hat{b}(1, 2) | \psi_j \psi_i \rangle \\ &= \frac{1}{2} \sum_{ij}^N \iint \psi_i^*(\mathbf{x}_1) \psi_j^*(\mathbf{x}_2) \hat{b}(\mathbf{x}_1, \mathbf{x}_2) \psi_i(\mathbf{x}_1) \psi_j(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \\ &\quad - \frac{1}{2} \sum_{ij}^N \iint \psi_i^*(\mathbf{x}_1) \psi_j^*(\mathbf{x}_2) \hat{b}(\mathbf{x}_1, \mathbf{x}_2) \psi_j(\mathbf{x}_1) \psi_i(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \end{aligned}$$

21/99

The Hartree-Fock method

# Foundations

The basic assumption in Hartree-Fock (HF) theory is that a single Slater determinant is a good approximation of the wave function.

- The ground-state wavefunction and energy are obtained with the variational principle.
- Lagrange multipliers enforce orthonormal orbitals in the Slater determinant.
- The same mathematical tools are used in the Kohn-Sham formalism in DFT.
- HF concepts such as ‘exchange’ and (lack of) ‘correlation’ are also important in DFT.

23/99

## The Hartree-Fock Hamiltonian

General expression of the electronic Hamiltonian

$$\hat{H}_e = \sum_i^N \hat{h}(i) + \frac{1}{2} \sum_{i \neq j}^N \hat{V}_{ee}(i, j)$$

$$\hat{h}(i) = -\frac{1}{2} \nabla_i^2 + \hat{V}_{ext}(i)$$

$$\hat{V}_{ee}(i, j) = \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$$

Expectation values with Slater determinant

$$\langle \Psi | \hat{h}(i) | \Psi \rangle = -\frac{1}{2} \langle \psi_i | \nabla_1^2 | \psi_i \rangle + \langle \psi_i | \hat{V}_{ext}(1) | \psi_i \rangle$$

$$\langle \Psi | \hat{V}_{ee}(i, j) | \Psi \rangle = \langle \psi_i \psi_j | \hat{V}_{ee}(1, 2) | \psi_i \psi_j \rangle - \langle \psi_i \psi_j | \hat{V}_{ee}(1, 2) | \psi_j \psi_i \rangle$$

# The Variational principle and Hartree-Fock (1)

The ground state minimizes the energy:

$$E_{\text{HF}} = \sum_i^N \langle \Psi | \hat{h}(i) | \Psi \rangle + \frac{1}{2} \sum_{i \neq j}^N \langle \Psi | \hat{V}_{\text{ee}}(i, j) | \Psi \rangle$$

under the following constraints

$$\langle \psi_i | \psi_j \rangle = \delta_{ij} \quad \forall i, j.$$

The constrained minimum of  $E_{\text{HF}}$  is a stationary point of the Lagrangian:

$$L_{\text{HF}} = E_{\text{HF}} - \sum_{ij}^N (\langle \psi_i | \psi_j \rangle - \delta_{ij}) \epsilon_{ji}.$$

25/99

# The Variational principle and Hartree-Fock (2)

The functional derivative of the Lagrangian towards an orbital is:

$$\frac{\delta L_{\text{HF}}}{\delta \psi_i^*(\mathbf{x})} = \frac{\delta E_{\text{HF}}}{\delta \psi_i^*(\mathbf{x})} - \sum_j^N \psi_j(\mathbf{x}) \epsilon_{ji}$$

Setting these derivatives to zero yields the Roothaan equations.

$$\frac{\delta E_{\text{HF}}}{\delta \psi_i^*(\mathbf{x})} = \sum_j^N \psi_j(\mathbf{x}) \epsilon_{ji}$$

## The Variational principle and Hartree-Fock (3)

The derivative of the Hartree-Fock energy becomes

$$\begin{aligned} \frac{\delta E_{\text{HF}}}{\delta \psi_i^*(\mathbf{x})} = & -\frac{1}{2} \nabla^2 \psi_i(\mathbf{x}) + V_{\text{ext}}(\mathbf{r}) \psi_i(\mathbf{x}) \\ & + \sum_j^N \int \psi_j^*(\mathbf{x}_2) \frac{1}{|\mathbf{r} - \mathbf{r}_2|} \psi_i(\mathbf{x}) \psi_j(\mathbf{x}_2) d\mathbf{x}_2 \\ & - \sum_j^N \int \psi_j^*(\mathbf{x}_2) \frac{1}{|\mathbf{r} - \mathbf{r}_2|} \psi_i(\mathbf{x}_2) \psi_j(\mathbf{x}) d\mathbf{x}_2 \end{aligned}$$

27 / 99

## The Variational principle and Hartree-Fock (4)

The last two terms can be written with operators

$$\frac{\delta E_{\text{HF}}}{\delta \psi_i^*(\mathbf{x})} = -\frac{1}{2} \nabla^2 \psi_i(\mathbf{x}) + V_{\text{ext}}(\mathbf{r}) \psi_i(\mathbf{x}) + \hat{J} \psi_i(\mathbf{x}) - \hat{K} \psi_i(\mathbf{x})$$

With the following definition for the Coulomb or Hartree ( $\hat{J}$ ) and exchange ( $\hat{K}$ ) one-body operators

$$\hat{J}f(\mathbf{x}) = \sum_j^N \int \psi_j^*(\mathbf{x}_2) \frac{1}{|\mathbf{r} - \mathbf{r}_2|} \psi_j(\mathbf{x}_2) f(\mathbf{x}) d\mathbf{x}_2$$

$$\hat{K}f(\mathbf{x}) = \sum_j^N \int \psi_j^*(\mathbf{x}_2) \frac{1}{|\mathbf{r} - \mathbf{r}_2|} f(\mathbf{x}_2) \psi_j(\mathbf{x}) d\mathbf{x}_2$$

The sum of the four terms is called the Fock operator

$$\frac{\delta E_{\text{HF}}}{\delta \psi_i^*(\mathbf{x})} = \hat{F} \psi_i(\mathbf{x})$$

# The Variational principle and Hartree-Fock (5)

The Roothaan equations in operator notation:

$$\hat{F}\psi_i(\mathbf{x}) = \sum_j \psi_j(\mathbf{x})\epsilon_{ji}$$

One can show that:

- the matrix with Lagrange multipliers is Hermitian ( $\epsilon_{ji} = \epsilon_{ij}^*$ ),
- a unitary transformation of the orbitals,  $\psi_j$ , only changes the phase of the N-body wavefunction  $\Psi$  and does not change any expectation value.

Therefore, one can always apply a unitary transformation to the orbitals such that all off-diagonal Lagrange multipliers become zero:

$$\hat{F}\psi_i(\mathbf{x}) = \epsilon_{ii}\psi_i(\mathbf{x}) = \epsilon_i\psi_i(\mathbf{x})$$

This is a classical analytical eigenproblem. The eigenvalues are called the orbital energies.

29/99

# Hartree-Fock in a finite basis (1)

The solution of the eigenproblem is implemented on computers by introducing a set of basis functions  $b_k$ . Each orbital is expanded in this basis:

$$\psi_i(\mathbf{x}) = \sum_k^{N_b} [C]_{ki} b_k(\mathbf{x})$$

where  $N_b$  is the number of basis functions. No more than  $N_b$  orthogonal orbitals can be constructed. In these notes  $C$  is a square matrix.

The first  $N$  orbitals (present in Slater determinant) are occupied. The remaining  $N_b - N$  orbitals are called *virtual* orbitals and are *not* used in the Slater determinant.

- **Occupied:**  $[O]_{ii} = n_i = 1 \quad \forall \quad i \in 1 \dots N$
- **Virtual:**  $[O]_{ii} = n_i = 0 \quad \forall \quad i \in N + 1 \dots N_b$

30/99

## Hartree-Fock in a finite basis (2)

These basis functions may be orthonormal. If not, the overlap matrix differs from the identity matrix:

$$[S]_{k\ell} = \int b_k^*(\mathbf{x}) b_\ell(\mathbf{x}) d\mathbf{x} \neq \delta_{k\ell}$$

The orthogonality of the orbitals is expressed in matrix notation as:

$$C^\dagger S C = I$$

31/99

## Hartree-Fock in a finite basis (3)

The kinetic energy in terms of expansion coefficients:

$$T = \sum_i^N \sum_{k\ell}^{N_b} \int [C]_{ki}^* b_k^*(\mathbf{x}) \left( -\frac{1}{2} \nabla^2 \right) [C]_{\ell i} b_\ell(\mathbf{x}) d\mathbf{x}$$

Using the following two matrix definitions

$$[D]_{\ell k} = \sum_i^{N_b} [C]_{\ell i} [O]_{ii} [C]_{ki}^* \quad [\mathcal{T}]_{k\ell} = \int b_k^*(\mathbf{x}) \left( -\frac{1}{2} \nabla^2 \right) b_\ell(\mathbf{x}) d\mathbf{x}$$

$$D = C O C^\dagger$$

where  $D$  is called the density matrix (Hermitian),  $O$  is a diagonal matrix with occupation numbers (0 or 1). The kinetic energy is now written compactly as

$$T = \sum_{k\ell} [D]_{\ell k} [\mathcal{T}]_{k\ell} = \sum_\ell [D\mathcal{T}]_{\ell\ell} = \text{tr}(D\mathcal{T})$$

## Hartree-Fock in a finite basis (4)

The interaction with the external potential is very similar:

$$V_{\text{ext}} = \text{tr}(D^T \mathcal{V}_{\text{ext}})$$

with

$$[\mathcal{V}_{\text{ext}}]_{k\ell} = \int b_k^*(\mathbf{x}) V_{\text{ext}}(\mathbf{r}) b_\ell(\mathbf{x}) d\mathbf{x}$$

These are the nuclear attraction integrals (NAI).

33/99

## Hartree-Fock in a finite basis (5)

The electron-electron interaction is again similar, but a bit more complicated

$$V_{\text{ee}} = \frac{1}{2} \sum_{ij}^N \sum_{k\ell mn}^{N_b} \iint d\mathbf{x}_1 d\mathbf{x}_2 \left( \begin{aligned} & C_{ki}^* b_k^*(\mathbf{x}_1) C_{\ell j}^* b_\ell^*(\mathbf{x}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} C_{mi} b_m(\mathbf{x}_1) C_{nj} b_n(\mathbf{x}_2) \\ & - C_{ki}^* b_k^*(\mathbf{x}_1) C_{\ell j}^* b_\ell^*(\mathbf{x}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} C_{mj} b_m(\mathbf{x}_1) C_{ni} b_n(\mathbf{x}_2) \end{aligned} \right)$$

The integrals over the basis functions can be isolated and take the following form: (electron repulsion integrals, ERI)

$$[\mathcal{V}_{\text{ee}}]_{k\ell mn} = \int b_k^*(\mathbf{x}_1) b_\ell^*(\mathbf{x}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} b_m(\mathbf{x}_1) b_n(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2$$

## Hartree-Fock in a finite basis (6)

Using the ERI matrix and the density matrix, one may rewrite the electron-electron interaction in a compact form:

$$V_{ee} = \frac{1}{2} \sum_{k\ell mn}^{N_b} ([D]_{km} [\mathcal{V}_{ee}]_{k\ell mn} [D]_{\ell n} - [D]_{km} [\mathcal{V}_{ee}]_{k\ell nm} [D]_{\ell n})$$

This can not be simply cast in conventional algebraic notation because of the four-index matrix.

Note that the HF energy is simply quadratic in the density matrix

35/99

## Varitional principle for Hartree-Fock in a finite basis (1)

The ground state in the finite basis minimizes the energy:

$$E_{HF} = T(C) + V_{ext}(C) + V_{ee}(C)$$

under the following orthonormality constraints for the orbitals:

$$C^\dagger S C = I$$

Hence, the ground state is a stationary point of the Lagrangian:

$$L_{HF} = E_{HF} - \sum_{ij}^{N_b} [C^\dagger S C - I]_{ij} [\epsilon]_{ji}.$$

## Varitional principle for Hartree-Fock in a finite basis (2)

The stationary point is found by setting the derivative towards the expansion coefficients to zero:

$$\frac{\partial L_{\text{HF}}}{\partial [C]_{ki}^*} = \frac{\partial E_{\text{HF}}}{\partial [C]_{ki}^*} - \sum_j^{N_b} [SC]_{kj} [\epsilon]_{ji} = 0$$

Again, a unitary transformation of the orbitals does not change the solution. It allows us to choose the matrix of Lagrange multipliers,  $\epsilon$ , to be diagonal:

$$\frac{\partial L_{\text{HF}}}{\partial [C]_{ki}^*} = \frac{\partial E_{\text{HF}}}{\partial [C]_{ki}^*} - [SC]_{ki} [\epsilon]_{ii} = 0$$

37/99

## Varitional principle for Hartree-Fock in a finite basis (3)

Using the chain rule, the derivative towards the density matrix is inserted:

$$\frac{\partial E_{\text{HF}}}{\partial [C]_{ki}^*} = \sum_m^{N_b} \frac{\partial E_{\text{HF}}}{\partial [D]_{mk}} [C]_{mi}$$

By defining the Fock matrix as follows

$$[F]_{km} = \frac{\partial E_{\text{HF}}}{\partial [D]_{mk}}$$

the Roothaan equations take the following familiar form in matrix notation:

$$FC = SC\epsilon$$

This is a generalized (due to  $S$ ) algebraic eigenproblem with the same form as the general eigenproblem.

38/99

## Varitional principle for Hartree-Fock in a finite basis (4)

The last step is to write out the Fock matrix. The HF energy is quadratic in the density matrix. Thus the Fock matrix is linear in the Density matrix:

$$[F]_{km} = [\mathcal{T}]_{km} + [\mathcal{V}_{\text{ext}}]_{km} + \sum_{\ell n} [\mathcal{V}_{\text{ee}}]_{k\ell mn} [D]_{\ell n} - \sum_{\ell n} [\mathcal{V}_{\text{ee}}]_{k\ell nm} [D]_{\ell n}$$

The Fock matrix depends on the solution! The last two terms are called the Coulomb or Hartree matrix ( $\mathcal{J}$ ) and the exchange matrix ( $\mathcal{K}$ ), respectively:

$$[F]_{km} = [\mathcal{T}]_{km} + [\mathcal{V}_{\text{ext}}]_{km} + [\mathcal{J}]_{km} - [\mathcal{K}]_{km}$$

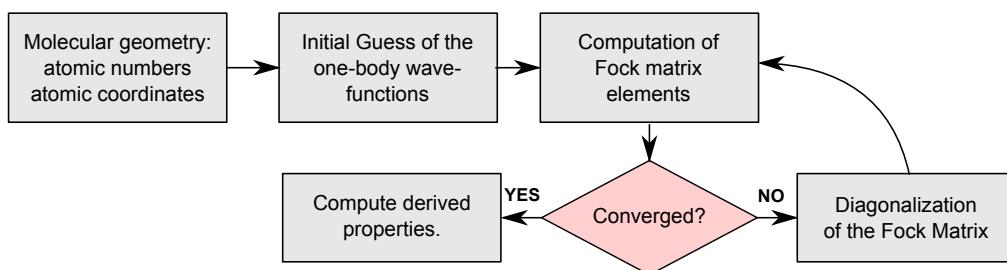
Note similarities & differences with the electronic hamiltonian!

39/99

## Self-consistent Field (SCF)

The Rothean equations must actually be solved iteratively because the Fock matrix directly depends on the wavefunctions. The SCF algorithm finds a Fock matrix and a set of consistent orbitals such that:

- ① the wavefunctions are eigenvectors of the Fock matrix and
- ② the Fock matrix is obtained from the wavefunctions.



The naive approach may lead to bi-stable solutions.

40/99

## Relation between Lagrange multipliers and $E_{\text{HF}}$

One can not simply add up the orbital energies to obtain the Hartree-Fock energy. The following relations are valid:

$$\begin{aligned} E_{\text{HF}} &= \sum_i^N \langle \hat{h}(i) \rangle + \frac{1}{2} \sum_{i \neq j}^N \langle \hat{V}_{\text{ee}}(i, j) \rangle \\ &= \sum_i^N \epsilon_i - \frac{1}{2} \sum_{i \neq j}^N \langle \hat{V}_{\text{ee}}(i, j) \rangle \end{aligned}$$

Interpretation of  $\epsilon_i$ : Koopman's Theorem

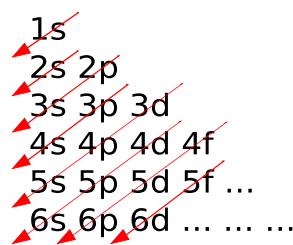
$$\epsilon_m \approx -I_m$$

where  $I_m$  is the ionization potential of the  $m$ 'th electron.

41/99

## Aufbau principle and excited states

- When searching for the ground state, the  $N$  lowest-energy orbitals are systematically occupied. (Aufbau principle.) for atomic orbitals:



This is a Hartree-Fock result, here without proof.

- Excited states can be constructed by violation of the Aufbau principle.

With  $N_b$  basis functions 1  $N$ -electron ground-state Slater determinant and  $\frac{N_b!}{N!(N_b-N)!} - 1$  excited-state Slater determinants can be constructed.

42/99

## Treatment of $e^-$ spin

Three variations of Hartree-Fock for electronic systems. Each one assumes  $\psi(\mathbf{x}) = \sigma(s)\phi(\mathbf{r})$ .

## Restricted closed shell HF: $N$ is even (RHF)

$$\Psi = \det[\alpha\phi_1, \dots, \alpha\phi_{N/2}, \beta\phi_1, \dots, \beta\phi_{N/2}](\mathbf{x}_1, \dots, \mathbf{x}_N)$$

## Restricted open shell HF: $N_\alpha > N_\beta$ (ROHF)

$$\Psi = \det[\alpha\phi_1, \dots, \alpha\phi_{N_\alpha}, \beta\phi_1, \dots, \beta\phi_{N_\beta}](\mathbf{x}_1, \dots, \mathbf{x}_N)$$

### Unrestricted HF: $N_\alpha > N_\beta$ (UHF)

$$\Psi = \det[\alpha\phi_1^\alpha, \dots, \alpha\phi_{N_\alpha}^\alpha, \beta\phi_1^\beta, \dots, \beta\phi_{N_\beta}^\beta](\mathbf{x}_1, \dots, \mathbf{x}_N)$$

Each ansatz has its own Slater-Condon-like rules.

43/99

## Restricted closed-shell HF (1)

## One-body operator for N particles **without spin-dependence**:

$$\hat{A} = \sum_{i=1}^N \hat{a}(i)$$

The expectation value becomes:

$$\langle \Psi | \hat{A} | \Psi \rangle = 2 \sum_i^{N/2} \langle \phi_i | \hat{a}(1) | \phi_i \rangle = 2 \sum_i^{N/2} \int \phi_i^*(\mathbf{r}_1) \hat{a}(\mathbf{r}_1) \phi_i(\mathbf{r}_1) d\mathbf{r}_1$$

## Restricted closed-shell HF (2)

Two-body operator for N particles **without spin-dependence**:

$$\hat{B} = \frac{1}{2} \sum_{i \neq j}^N \hat{b}(i, j)$$

The expectation value becomes:

$$\begin{aligned} \langle \Psi | \hat{B} | \Psi \rangle &= \sum_{ij}^{N/2} \left[ 2\langle \phi_i \phi_j | \hat{b}(1, 2) | \phi_i \phi_j \rangle - \langle \phi_i \phi_j | \hat{b}(1, 2) | \phi_j \phi_i \rangle \right] \\ &= 2 \sum_{ij}^{N/2} \iint \phi_i^*(\mathbf{r}_1) \phi_j^*(\mathbf{r}_2) \hat{b}(\mathbf{r}_1, \mathbf{r}_2) \phi_i(\mathbf{r}_1) \phi_j(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \\ &\quad - \sum_{ij}^{N/2} \iint \phi_i^*(\mathbf{r}_1) \phi_j^*(\mathbf{r}_2) \hat{b}(\mathbf{r}_1, \mathbf{r}_2) \phi_j(\mathbf{r}_1) \phi_i(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \end{aligned}$$

45/99

## Unrestricted open-shell HF (1)

One-body operator for N particles **without spin-dependence**:

$$\hat{A} = \sum_i^{N_\alpha} \hat{a}(i) + \sum_i^{N_\beta} \hat{a}(i)$$

The expectation value becomes:

$$\langle \Psi | \hat{A} | \Psi \rangle = \sum_{i=1}^{N_\alpha} \langle \phi_i^\alpha | \hat{a}(1) | \phi_i^\alpha \rangle + \sum_{i=1}^{N_\beta} \langle \phi_i^\beta | \hat{a}(1) | \phi_i^\beta \rangle$$

## Unrestricted open-shell HF (2)

Two-body operator for N particles **without spin-dependence**:

$$\hat{B} = \frac{1}{2} \sum_{i \neq j}^N \hat{b}(i, j)$$

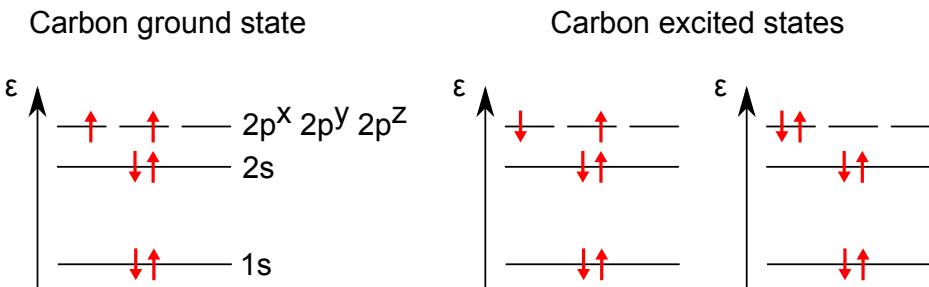
The expectation value becomes:

$$\begin{aligned} \langle \Psi | \hat{B} | \Psi \rangle = & \frac{1}{2} \sum_{ij}^{N_\alpha} \left[ \langle \phi_i^\alpha \phi_j^\alpha | \hat{b}(1, 2) | \phi_i^\alpha \phi_j^\alpha \rangle - \langle \phi_i^\alpha \phi_j^\alpha | \hat{b}(1, 2) | \phi_j^\alpha \phi_i^\alpha \rangle \right] \\ & + \frac{1}{2} \sum_{ij}^{N_\beta} \left[ \langle \phi_i^\beta \phi_j^\beta | \hat{b}(1, 2) | \phi_i^\beta \phi_j^\beta \rangle - \langle \phi_i^\beta \phi_j^\beta | \hat{b}(1, 2) | \phi_j^\beta \phi_i^\beta \rangle \right] \\ & + \sum_i^{N_\alpha} \sum_j^{N_\beta} \langle \phi_i^\alpha \phi_j^\beta | \hat{b}(1, 2) | \phi_i^\alpha \phi_j^\beta \rangle \end{aligned}$$

47/99

## Hund's rule

**Rule:** When two spatial orbitals are degenerate ( $\phi_i(\mathbf{r}) \neq \phi_j(\mathbf{r})$  and  $\epsilon_i = \epsilon_j$ ), the orbitals are first filled with aligned spins. Example for carbon:



This is a Hartree-Fock result, here without proof.  
(See ptable.com for more examples.)

## Density and density matrix.

Two important concepts in DFT and electronic structure theory in general.

49/99

Definitions and properties

## N-Particle density (matrix) (1)

N-Particle density:

$$n_N(\mathbf{x}_1, \dots, \mathbf{x}_N) = \Psi^*(\mathbf{x}_1, \dots, \mathbf{x}_N) \Psi(\mathbf{x}_1, \dots, \mathbf{x}_N)$$

N-Particle density matrix (NDM) = a generalization:

$$\gamma_N(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{x}'_1, \dots, \mathbf{x}'_N) = \Psi^*(\mathbf{x}_1, \dots, \mathbf{x}_N) \Psi(\mathbf{x}'_1, \dots, \mathbf{x}'_N)$$

This is not an algebraic matrix!

## N-Particle density (matrix) (2)

Any expectation value can be computed from the N-Particle density matrix. Start from a generic N-body operator:

$$\hat{A}(\mathbf{x}_1, \dots, \mathbf{x}_N)$$

The expectation value:

$$\begin{aligned} \langle A \rangle &= \int \Psi^*(\mathbf{x}_1, \dots, \mathbf{x}_N) \hat{A}(\mathbf{x}_1, \dots, \mathbf{x}_N) \Psi(\mathbf{x}_1, \dots, \mathbf{x}_N) d\mathbf{x}_1 \dots d\mathbf{x}_N \\ &= \int [\hat{A}(\mathbf{x}'_1, \dots, \mathbf{x}'_N) \Psi^*(\mathbf{x}_1, \dots, \mathbf{x}_N) \Psi(\mathbf{x}'_1, \dots, \mathbf{x}'_N)]_{\mathbf{x}'_1 = \mathbf{x}_1, \dots, \mathbf{x}'_N = \mathbf{x}_N} d\mathbf{x}_1 \dots d\mathbf{x}_N \\ &= \int [\hat{A}(\mathbf{x}'_1, \dots, \mathbf{x}'_N) \gamma_N(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{x}'_1, \dots, \mathbf{x}'_N)]_{\mathbf{x}'_1 = \mathbf{x}_1, \dots, \mathbf{x}'_N = \mathbf{x}_N} d\mathbf{x}_1 \dots d\mathbf{x}_N \end{aligned}$$

Hence, the NDM contains all information!

## 1-Particle reduced density (matrix)

1-Particle reduced density (a.k.a. *the density*):

$$\begin{aligned} n(\mathbf{x}_a) = n_1(\mathbf{x}_a) &= \left\langle \Psi \left| \sum_i^N \delta(\mathbf{x}_a - \mathbf{x}_i) \right| \Psi \right\rangle \\ &= N \int n_N(\mathbf{x}_a, \mathbf{x}_2, \dots, \mathbf{x}_N) d\mathbf{x}_2 \dots d\mathbf{x}_N \end{aligned}$$

1-Particle reduced density matrix (a.k.a. *the density matrix*):

$$\gamma(\mathbf{x}_a, \mathbf{x}'_a) = \gamma_1(\mathbf{x}_a, \mathbf{x}'_a) = N \int \gamma_N(\mathbf{x}_a, \mathbf{x}_2, \dots, \mathbf{x}_N, \mathbf{x}'_a, \mathbf{x}_2, \dots, \mathbf{x}_N) d\mathbf{x}_2 \dots d\mathbf{x}_N$$

Without proof: any one-body expectation value can be derived from  $\gamma_1$

## Definitions and properties

## 1-Particle reduced density (matrix) — $\Psi = \text{Slater det. (1)}$

1-Particle reduced density (a.k.a. *the density*) with Slater-Condon rules:

$$n_1^S(\mathbf{x}_a) = \left\langle \Psi \left| \sum_i^N \delta(\mathbf{x}_a - \mathbf{x}_i) \right| \Psi \right\rangle$$

$$= \sum_i^N \langle \psi_i | \delta(\mathbf{x}_a - \mathbf{x}_i) | \psi_i \rangle = \sum_i^N |\psi_i(\mathbf{x}_a)|^2$$

1-Particle reduced density matrix (a.k.a. *the density matrix*) without proof:

$$\gamma_1^S(\mathbf{x}_a, \mathbf{x}'_a) = \sum_i^N \psi_i^*(\mathbf{x}_a) \psi_i(\mathbf{x}'_a)$$

53/99

## Definitions and properties

## 1-Particle reduced density (matrix) — $\Psi = \text{Slater det. (2)}$

1-Particle reduced density (a.k.a. *the density* or *1RD*) with finite basis and Slater-Condon rules:

$$n_1^S(\mathbf{x}_a) = \sum_i^N \sum_{k\ell}^{N_b} [C]_{ki}^* [C]_{li} b_k^*(\mathbf{x}_a) b_\ell(\mathbf{x}_a)$$

$$= \sum_{k\ell}^{N_b} [D]_{kl} b_k^*(\mathbf{x}_a) b_\ell(\mathbf{x}_a)$$

1-Particle reduced density matrix (a.k.a. *the density matrix* or *1RDM*) with finite basis:

$$\begin{aligned}\gamma_1^S(\mathbf{x}_a, \mathbf{x}'_a) &= \sum_i^N \sum_{k\ell}^{N_b} [C]_{ki}^* [C]_{li} b_i^k(\mathbf{x}_a) b_\ell(\mathbf{x}'_a) \\ &= \sum_{k\ell}^{N_b} [D]_{kl} b_k^*(\mathbf{x}_a) b_\ell(\mathbf{x}'_a)\end{aligned}$$

# 1-Particle reduced density (matrix) — $\Psi = \text{Slater det.}$ (3)

The density matrix fully defines the Slater determinant. We start from the definition of the algebraic density matrix:

$$D = COC^\dagger$$

Then right-multiply by  $SC$ , left-multiply by  $S$  and use the orthogonality  $C^\dagger SC = I$ :

$$SDSC = SCO$$

This is also a generalized eigenvalue problem:

- One can recover some set of orbitals from the density matrix. These are called the natural orbitals.
- Due to degeneracy of occupation numbers, the natural orbitals are only known up to a unitary transformation of the occupied and virtual orbitals.
- These unitary transformations are not relevant for Slater determinant.

55/99

# 2-Particle reduced density (matrix) (1)

2-Particle reduced density (2RD):

$$\begin{aligned} n_2(\mathbf{x}_a, \mathbf{x}_b) &= \left\langle \Psi \left| \sum_{i=1}^N \sum_{j=i+1}^N \delta(\mathbf{x}_a - \mathbf{x}_i) \delta(\mathbf{x}_b - \mathbf{x}_j) \right| \Psi \right\rangle \\ &= \frac{N(N-1)}{2} \int n_N(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_3, \dots, \mathbf{x}_N) d\mathbf{x}_3 \dots d\mathbf{x}_N \end{aligned}$$

2-Particle reduced density matrix (2RDM):

$$\gamma_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}'_a, \mathbf{x}'_b) = \frac{N(N-1)}{2} \int \gamma_N(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_3, \dots, \mathbf{x}_N, \mathbf{x}'_a, \mathbf{x}'_b, \mathbf{x}_3, \dots, \mathbf{x}_N) d\mathbf{x}_3 \dots d\mathbf{x}_N$$

Without proof: any two-body expectation value can be derived from  $\gamma_2$ .

## 2-Particle reduced density (matrix) (2)

Without proof: any two-body expectation value can be derived from  $\gamma_2$ .

For example, the repulsion between the electrons can be written as:

$$V_{ee} = \int \frac{n_2(\mathbf{x}_1, \mathbf{x}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{x}_1 d\mathbf{x}_2$$

This is straightforward generalization of the electrostatic interaction in a classical charge density. The uncorrelated 2-particle density,  $n_1(\mathbf{x}_1)n_1(\mathbf{x}_2)/2$  is replaced by a correlated probability density:  $n_2(\mathbf{x}_1, \mathbf{x}_2)$ .

## The two-particle density — $\Psi = \text{Slater det.}$ (1)

2-Particle reduced density with Slater-Condon rules:

$$\begin{aligned}
 n_2^S(\mathbf{x}_a, \mathbf{x}_b) &= \frac{1}{2} \left\langle \Psi \left| \sum_{i \neq j}^N \delta(\mathbf{x}_a - \mathbf{x}_i) \delta(\mathbf{x}_b - \mathbf{x}_j) \right| \Psi \right\rangle \\
 &= \frac{1}{2} \sum_{ij}^N [\langle \psi_i \psi_j | \delta(\mathbf{x}_a - \mathbf{x}_1) \delta(\mathbf{x}_b - \mathbf{x}_2) | \psi_i \psi_j \rangle \\
 &\quad - \langle \psi_i \psi_j | \delta(\mathbf{x}_a - \mathbf{x}_1) \delta(\mathbf{x}_b - \mathbf{x}_2) | \psi_j \psi_i \rangle] \\
 &= \frac{1}{2} \sum_{ij}^N [|\psi_i(\mathbf{x}_a)|^2 |\psi_j(\mathbf{x}_a)|^2 - \psi_i^*(\mathbf{x}_a) \psi_i(\mathbf{x}_b) \psi_j^*(\mathbf{x}_b) \psi_j(\mathbf{x}_a)] \\
 &= \frac{1}{2} n_1^S(\mathbf{x}_a) n_1^S(\mathbf{x}_b) - \frac{1}{2} \left| \sum_i^N \psi_i^*(\mathbf{x}_a) \psi_i(\mathbf{x}_b) \right|^2
 \end{aligned}$$

# The two-particle density — $\Psi = \text{Slater det. (2)}$

2-Particle reduced density with Slater-Condon rules:

$$n_2^S(\mathbf{x}_a, \mathbf{x}_b) = \frac{n_1^S(\mathbf{x}_a)n_1^S(\mathbf{x}_b) - |\gamma_1^S(\mathbf{x}_a, \mathbf{x}_b)|^2}{2}$$

2-Particle reduced density matrix without proof:

$$\gamma_2^S(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}'_a, \mathbf{x}'_b) = \frac{1}{2} \begin{vmatrix} \gamma_1^S(\mathbf{x}_a, \mathbf{x}'_a) & \gamma_1^S(\mathbf{x}_a, \mathbf{x}'_b) \\ \gamma_1^S(\mathbf{x}_b, \mathbf{x}'_a) & \gamma_1^S(\mathbf{x}_b, \mathbf{x}'_b) \end{vmatrix}$$

## Final remarks

- N-body wavefunction and NDM are equivalent.
- For Slater determinant wavefunction:
  - Slater determinant and 1RDM are equivalent.
  - 1RDM describes deviations from classical charge distribution.
  - Hartree-Fock can also be defined by 1RDM ansatz instead of Slater determinant ansatz. The HF 1RDM has eigenvalues (occupation numbers) 0 or 1.
  - Hartree-Fock approximates the 2RDM. The residual is:
 
$$\gamma_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}'_a, \mathbf{x}'_b) - \frac{1}{2} \begin{vmatrix} \gamma_1^S(\mathbf{x}_a, \mathbf{x}'_a) & \gamma_1^S(\mathbf{x}_a, \mathbf{x}'_b) \\ \gamma_1^S(\mathbf{x}_b, \mathbf{x}'_a) & \gamma_1^S(\mathbf{x}_b, \mathbf{x}'_b) \end{vmatrix}$$
- One defines the spinless reduced density (matrix) by summing over the spin, e.g.  $n(\mathbf{r}) = \sum_{s=\alpha,\beta} n(\mathbf{r}, s) = \sum_{s=\alpha,\beta} n(\mathbf{x})$

The main shortcomings of the Hartree-Fock method.

61/99

## Correlation energy

- The Slater determinant is a limited expansion of the full N-body wavefunction.
- For a complete basis, a linear combination of Slater determinants is required. (Configuration interaction methods.)
- Hartree-Fock always overestimates the energy (variational principle)
- The missing part is called the correlation energy:

$$E = E_{\text{HF}} + E_{\text{cor}} \quad \text{with} \quad E_{\text{cor}} < 0$$

# Exchange and correlation (1)

- HF is a mean-field method: orbitals are 1-body wavefunctions for non-interacting particles.
- The Fock matrix is the 1-particle Hamiltonian.
- One electron only feels the mean field due to other electrons.
- Details of pair-wise interactions are averaged out.

63/99

# Exchange and correlation (2)

Physical interpretation:

- The HF electronic density is already correlated due to Pauli exclusion.

$$n_2(\mathbf{x}_a, \mathbf{x}_b) = \frac{1}{2}[n_1^S(\mathbf{x}_a)n_1^S(\mathbf{x}_b) - |\gamma_1^S(\mathbf{x}_a, \mathbf{x}_b)|^2]$$

However, this is the only form of correlation in Hartree-Fock, and is called *exchange*. The term *correlation* is actually reserved for all correlation effects that are absent in HF.

- Exchange effect = two electrons do not occupy the same orbital (Pauli exclusion).
- Correlation effect = two electrons want to avoid each other (Coulomb repulsion).

## Two types of correlation: static and dynamic

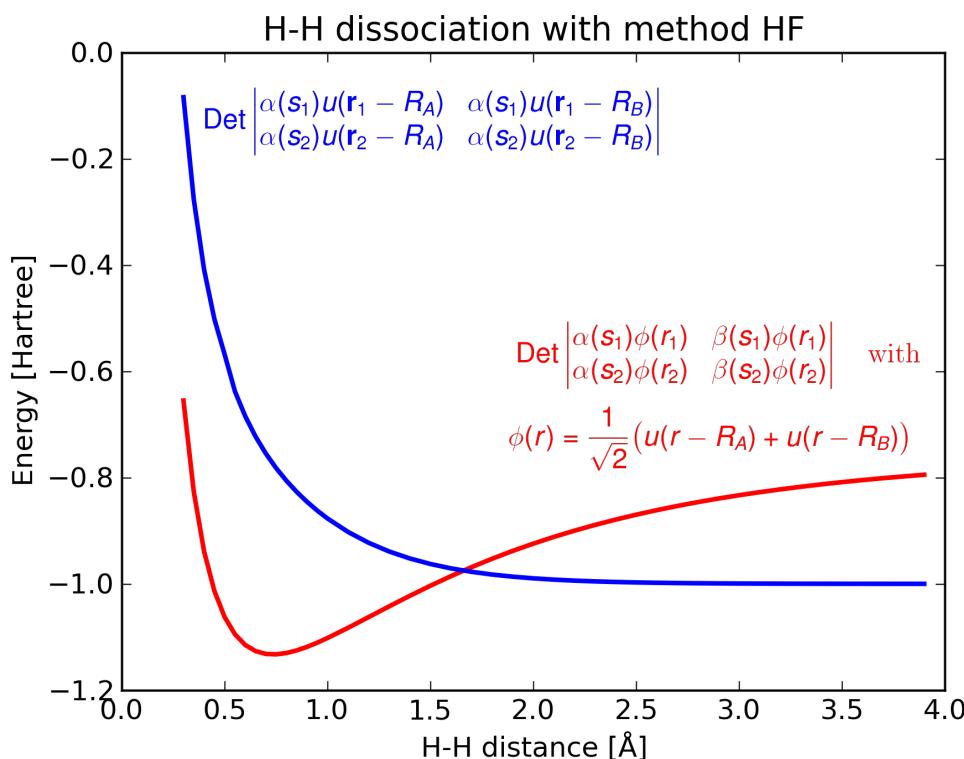
- **static:** 2 (or more) Slater determinants are needed
- **dynamic:** London dispersion interactions

Typical artifacts in computations:

- Covalent bonds are too short.
- Vibrational frequencies and polarizability are overestimated.
- Weak interactions are missing (often important in modern research).

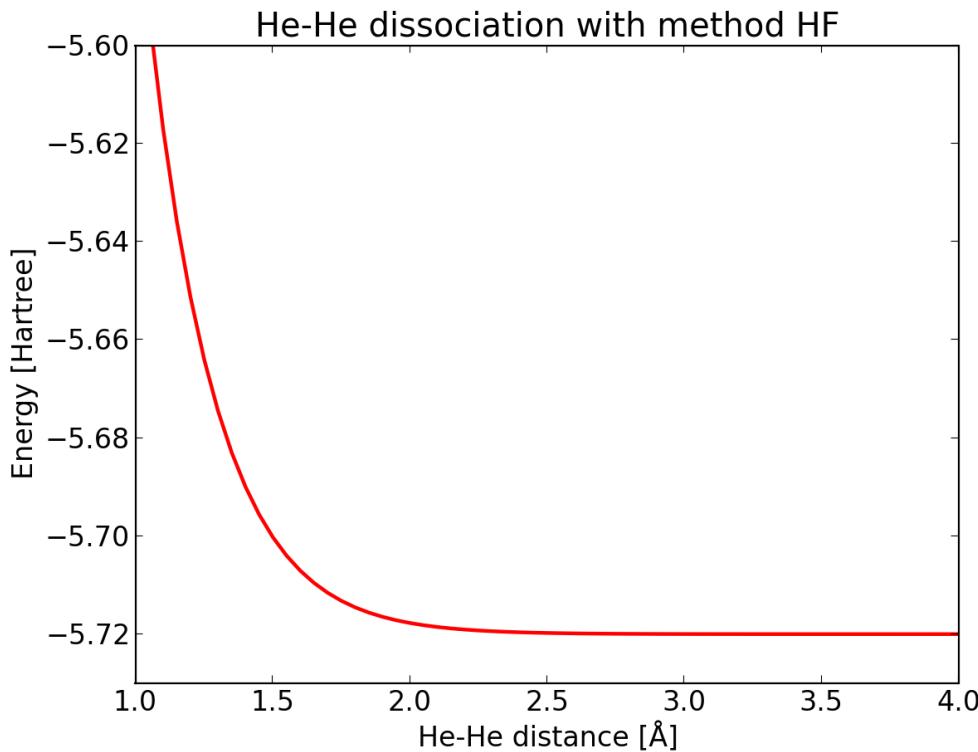
65/99

## Static correlation: H<sub>2</sub> dissociation



66/99

# Dynamic correlation: $\text{He}_2$ dissociation



67/99

# Electronic hamiltonian symmetries

The (non-relativistic) electronic hamiltonian commutes with:

- the spin operators  $\hat{S}_z$  and  $\hat{S}^2$
- any geometric symmetry of the nuclear coordinates.

The Fock operator does not necessarily have these symmetries!

- Only RHF and ROHF Fock matrices commute with  $\hat{S}^2$ . UHF is told to be sensitive to *spin contamination*.
- RHF, ROHF and UHF Fock matrices commute with  $\hat{S}_z$ .
- Only for a few elements, the HF atomic densities are spherically symmetrical.

Symmetry breaking

# The $\hat{S}_z$ operator

The Slater determinant is an eigenstate of  $\hat{S}_z$  when the orbitals are pure spin states:  $\psi(\mathbf{x}) = \sigma(s)\phi(\mathbf{r})$ . In practical implementations (RHF, ROHF, UHF), this is always the case.

$$\hat{S}_z = \sum_i^N \hat{s}_z(i)$$

Application of the one-body Slater-Condon rule:

$$\begin{aligned} \langle \Psi | \hat{S}_z | \Psi \rangle &= \sum_i^N \langle \psi_i | \hat{s}_z(1) | \psi_i \rangle \\ &= \frac{N_\alpha - N_\beta}{2} \end{aligned}$$

69/99

Symmetry breaking

# The $\hat{S}^2$ operator (1)

The treatment is more complicated for  $\hat{S}^2$ . We first rewrite this operator in a more convenient form:

$$\begin{aligned} \hat{S}^2 &= \hat{S}_x^2 + \hat{S}_y^2 + \hat{S}_z^2 \\ &= \hat{S}_z^2 + \hat{S}_z + \hat{S}_- \hat{S}_+ \end{aligned}$$

where we made use of

$$\begin{aligned} \hat{S}_+ &= \hat{S}_x + i\hat{S}_y \\ \hat{S}_- &= \hat{S}_x - i\hat{S}_y \\ [\hat{S}_x, \hat{S}_x] &= i\hat{S}_z \end{aligned}$$

## The $\hat{S}^2$ operator (2)

$\hat{S}^2$  is two-body operator that can be written in terms of one-body operators:

$$\hat{S}^2 = \left( \sum_i^N \hat{s}_z(i) \right)^2 + \sum_i^N \hat{s}_z(i) + \sum_i^N \hat{s}_-(i) \hat{s}_+(i) + \sum_{i \neq j}^N \hat{s}_-(i) \hat{s}_+(j)$$

The first three terms are trivial:  $\langle \Psi | \hat{S}_z | \Psi \rangle^2$ ,  $\langle \Psi | \hat{S}_z | \Psi \rangle$  and  $N_\beta$ . The fourth one requires some extra work.

71/99

## The $\hat{S}^2$ operator (3)

The fourth term ...

$$\begin{aligned} \left\langle \Psi \left| \sum_{i \neq j}^N \hat{s}_-(i) \hat{s}_+(j) \right| \Psi \right\rangle &= \sum_{ij} [\langle \psi_i \psi_j | \hat{s}_-(i) \hat{s}_+(j) | \psi_i \psi_j \rangle \\ &\quad - \langle \psi_i \psi_j | \hat{s}_-(i) \hat{s}_+(j) | \psi_j \psi_i \rangle] \end{aligned}$$

Only the exchange term remains, and only when  $\psi_i = \alpha \phi_i$  and  $\psi_j = \beta \phi_j$ . One finally obtains:

$$\begin{aligned} \left\langle \Psi \left| \sum_{i \neq j}^N \hat{s}_-(i) \hat{s}_+(j) \right| \Psi \right\rangle &= - \sum_{i \in \alpha} \sum_{j \in \beta} \langle \phi_i \phi_j | \phi_j \phi_i \rangle \\ &= - \sum_{i \in \alpha} \sum_{j \in \beta} \left| \int \phi_i(\mathbf{r}) \phi_j(\mathbf{r}) d\mathbf{r} \right|^2 \end{aligned}$$

# The $\hat{S}^2$ operator (4)

The final result:

$$\langle \hat{S}^2 \rangle = \langle \hat{S}_z \rangle^2 + \langle \hat{S}_z \rangle + N_\beta - \sum_{i \in \alpha} \sum_{j \in \beta} \left| \int \phi_i(\mathbf{r}) \phi_j(\mathbf{r}) d\mathbf{r} \right|^2$$

- In case of RHF and ROHF, the last two terms cancel.
- In the case of UHF, the last two terms may not cancel. This is called spin contamination and means that the HF solution breaks the  $\hat{S}^2$  symmetry of the original electronic hamiltonian.

73/99

# Four-center integrals

The Matrix elements of the operators  $\hat{J}$  and  $\hat{K}$  involve four-center integrals of the form

$$\int a(\mathbf{r}_1) b(\mathbf{r}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} c(\mathbf{r}_1) d(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2$$

The number of integrals scales with  $N_b^4$  where  $N_b$  is the number of basis functions.

- For  $\hat{J}$ , one may reduce  $N_b^4$  to  $N_b^3$  via the intermediate computation of the density and the Hartree potential.
- Proper screening of negligible integrals may be used to avoid the computation of small contributions to  $\hat{J}$  and  $\hat{K}$ . Scaling becomes  $N_b^2$ .
- One may also avoid the construction of the orbitals and work with sparse density matrices to represent the N-body wavefunction. Scaling becomes  $N_b$ .

74/99

## Density Functional Theory

75/99

Density as the basic quantity

## Foundations

The basic quantity in DFT is the density,  $n(\mathbf{r})$ .

Main advantages of DFT over HF:

- Inclusion of correlation effects in the Hamiltonian.
- Reduced formal complexity:  $n(\mathbf{r})$  versus  $\Psi(\mathbf{r}_1, \dots, \mathbf{r}_N)$ .

## Simple observations

Some HF expectation values are easily written as function of the density,  $n(\mathbf{r})$ :

$$\sum_i \langle \psi_i | \hat{V}_{\text{ext}} | \psi_i \rangle = V_{\text{ext}}[n(\mathbf{r})] = \int n(\mathbf{r}) V_{\text{ext}}(\mathbf{r}) d\mathbf{r}$$

$$\frac{1}{2} \sum_{ij} \langle \psi_i \psi_j | \hat{V}_{\text{ee}} | \psi_i \psi_j \rangle = J[n(\mathbf{r})] = \frac{1}{2} \iint \frac{n(\mathbf{r}_1) n(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2$$

Some are not:

$$-\frac{1}{2} \sum_i \langle \psi_i | \nabla_1^2 | \psi_i \rangle \stackrel{?}{=} T[n(\mathbf{r})]$$

$$-\frac{1}{2} \sum_{ij} \langle \psi_i \psi_j | \hat{V}_{\text{ee}} | \psi_j \psi_i \rangle \stackrel{?}{=} K[n(\mathbf{r})]$$

## First Hohenberg-Kohn Theorem (1)

**Theorem:**  $V_{\text{ext}}(\mathbf{r}) \Leftrightarrow n(\mathbf{r})$

- From left to right is trivial.
- From right to left is the Hohenberg-Kohn Theorem.

### Proof by contradiction

- Two **external potentials**  $V_{\text{ext}}(\mathbf{r})$  and  $V'_{\text{ext}}(\mathbf{r})$  that are **fundamentally different**, i.e.  $V_{\text{ext}}(\mathbf{r}) - V'_{\text{ext}}(\mathbf{r}) \neq \text{const}$
- Hamiltonians  $\hat{H}$  and  $\hat{H}'$  based on respective external potentials
- Each Hamiltonian has a non-degenerate ground state:  $\Psi$  and  $\Psi'$  with corresponding energies  $E_0$  and  $E'_0$ .
- $\Psi$  and  $\Psi'$  result in the **same density**,  $n(\mathbf{r})$ .

## First Hohenberg-Kohn Theorem (2)

- The variational principle requires that ...

$$E_0 < \langle \Psi' | \hat{H} | \Psi' \rangle = \langle \Psi' | \hat{H}' | \Psi' \rangle + \langle \Psi' | \hat{H} - \hat{H}' | \Psi' \rangle \\ = E'_0 + \int n(\mathbf{r}) [V_{\text{ext}}(\mathbf{r}) - V'_{\text{ext}}(\mathbf{r})] d\mathbf{r}$$

$$E'_0 < \langle \Psi | \hat{H}' | \Psi \rangle = \langle \Psi | \hat{H} | \Psi \rangle - \langle \Psi | \hat{H} - \hat{H}' | \Psi \rangle \\ = E_0 - \int n(\mathbf{r}) [V_{\text{ext}}(\mathbf{r}) - V'_{\text{ext}}(\mathbf{r})] d\mathbf{r}$$

Adding up both inequalities leads to a contradiction.

$$E_0 + E'_0 < E'_0 + E_0$$

⇒ Two different external potentials can not yield the same density.

79/99

## First Hohenberg-Kohn Theorem (3)

The first theorem implies the following relationships:

$$n(\mathbf{r}) \Leftrightarrow V_{\text{ext}}(\mathbf{r}) \Leftrightarrow \Psi(\mathbf{r}_1, \dots, \mathbf{r}_N) \Rightarrow \text{any property}$$

This implies that any property, e.g. the energy, is a functional of the density:

$$E = E[n(\mathbf{r})]$$

The energy functional may be written as a sum of terms corresponding to the electronic Hamiltonian:

$$E[n(\mathbf{r})] = T[n(\mathbf{r})] + V_{\text{ext}}[n(\mathbf{r})] + V_{\text{ee}}[n(\mathbf{r})] \\ = \int V_{\text{ext}}(\mathbf{r}) n(\mathbf{r}) d\mathbf{r} + F_{\text{HK}}[n(\mathbf{r})]$$

$$V_{\text{ee}}[n(\mathbf{r})] = J[n(\mathbf{r})] + \text{non-classical contribution}$$

# First Hohenberg-Kohn Theorem (4)

$$E[n(\mathbf{r})] = \int V_{\text{ext}}(\mathbf{r})n(\mathbf{r})d\mathbf{r} + F_{\text{HK}}[n(\mathbf{r})]$$

- The functional  $F_{\text{HK}}$  does not contain any reference to  $V_{\text{ext}}$  and is therefore *universal* for all electronic many-body problems.
- Once  $F_{\text{HK}}$  is known exactly, DFT can be used to determine the ground state of any electronic many-body problem.
- The exact analytical form of  $F_{\text{HK}}$  remains elusive.
- Many approximations and DFT methods exist and are generally more accurate than Hartree-Fock.

81/99

# Second Hohenberg-Kohn Theorem (1)

**Short:** Variational principle in DFT

For any trial density,  $\tilde{n}(\mathbf{r}) \geq 0$  ( $\int \tilde{n}(\mathbf{r})d\mathbf{r} = N$ ), the energy in a given external potential,  $V_{\text{ext},0}$ , is equal to or larger than the ground state energy for that external potential:

$$E_0 \leq E[\tilde{n}(\mathbf{r})]$$

Can be derived from variational principle for wavefunctions:

- Non-degenerate ground state density:  $n_0(\mathbf{r}) \Rightarrow V_{\text{ext},0}(\mathbf{r}) \Rightarrow \Psi_0$   
 $\Rightarrow E_0 = \int V_{\text{ext},0}(\mathbf{r})n_0(\mathbf{r})d\mathbf{r} + F_{\text{HK}}[n_0(\mathbf{r})] = \langle \Psi_0 | \hat{H} | \Psi_0 \rangle$
- $V_{\text{ext}}$ -representable trial density:  $\tilde{n}(\mathbf{r}) \Rightarrow \tilde{V}_{\text{ext}}(\mathbf{r}) \Rightarrow \tilde{\Psi}$   
 $\Rightarrow E[\tilde{n}(\mathbf{r})] = \int V_{\text{ext},0}(\mathbf{r})\tilde{n}(\mathbf{r})d\mathbf{r} + F_{\text{HK}}[\tilde{n}(\mathbf{r})] = \langle \tilde{\Psi} | \hat{H} | \tilde{\Psi} \rangle > E_0$

## Second Hohenberg-Kohn Theorem (2)

Only  $V_{\text{ext}}$ -representable densities should be considered in variational HK-DFT.

### $V_{\text{ext}}$ -representable densities

- Densities that are generated by some external potential.
- No convenient mathematical prescription for such  $n(\mathbf{r})$ .

Variational calculus on a broader class of  $N$ -representable densities leads to the same ground state.

### $N$ -representable densities

- Densities that are derived from some antisymmetric wavefunction.
- Weaker condition
- $n(\mathbf{r}) \geq 0 \quad \int n(\mathbf{r}) d\mathbf{r} = N \quad \int |\nabla n^{\frac{1}{2}}(\mathbf{r})|^2 d\mathbf{r} < \infty$

83/99

## Chemical potential (1)

The ground state wavefunction is obtained with a constrained minimization.

The Lagrangian:

$$L[n] = E[n] - \mu \left( \int n(\mathbf{r}) d\mathbf{r} - N \right)$$

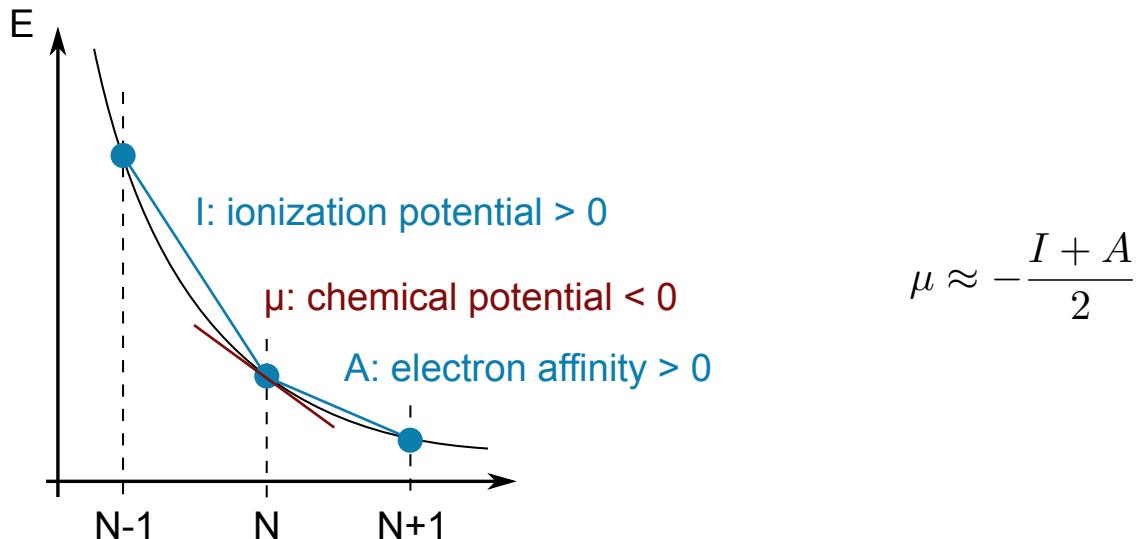
The derivative leads to:

$$\mu = \frac{\delta E[n]}{\delta n(\mathbf{r})} = V_{\text{ext}}(\mathbf{r}) + \frac{\delta F_{\text{HK}}[n]}{\delta n(\mathbf{r})}$$

At every point in space,  $\mathbf{r}$ , the derivative of the energy towards the density must be equal to the *chemical potential*,  $\mu$ .

Without proof:  $\mu = \left( \frac{dE}{dN} \right)_{V_{\text{ext}}}$

## Chemical potential (2)



85/99

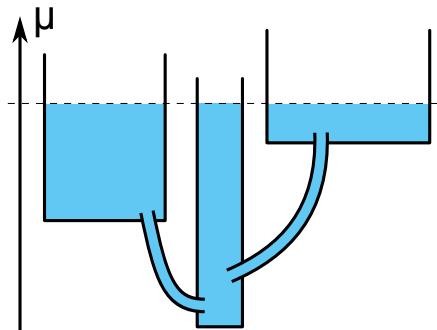
## Chemical potential (3)

In case of non-equilibrium densities,  $\tilde{n}$ , the chemical potential becomes a function:

$$\tilde{\mu}(\mathbf{r}) = \frac{\delta E[\tilde{n}]}{\delta \tilde{n}(\mathbf{r})}$$

By moving electrons from high to low chemical potential, one gets closer to the ground state density.

Cfr. communicating vessels:



# Local Density Approximation (LDA) (1)

In the case of a uniform electron gas, the kinetic and exchange energy **per particle** are known exactly:

- Thomas-Fermi:  $\mathcal{T}_{\text{TF}}[n] = \frac{3}{10}(3\pi^2)^{2/3}n^{2/3}$
- Dirac:  $\mathcal{E}_{\text{D},x}[n] = -\frac{3}{4}(3/\pi)^{1/3}n^{1/3}$

Similar approximate models exist for the correlation energy of a uniform electron gas:

- Wigner:  $\mathcal{E}_{\text{W},c}[n] = -\frac{1}{17.7 + 2.27 \cdot n^{-1/3}}$
- McWeeny:  $\mathcal{E}_{\text{M},c}[n] = -\frac{1}{9.652 + 2.946 \cdot n^{-1/3}}$

# Local Density Approximation (LDA) (2)

The properties of the uniform electron gas can be applied to finite systems with non-constant  $n(\mathbf{r})$  to construct an approximate Hohenberg-Kohn functional:

$$F_{\text{HK}}[n(\mathbf{r})] \approx J[n(\mathbf{r})] + \int n(\mathbf{r}) \times \left( \mathcal{T}_{\text{TF}}[n(\mathbf{r})] + \mathcal{E}_{\text{D},x}[n(\mathbf{r})] + \mathcal{E}_{\text{W},c}[n(\mathbf{r})] \right) d\mathbf{r}$$

This approach is called the Local Density Approximation.

Remarks:

- The largest error is due to the kinetic energy functional.
- LDA errors cancel each other out to a large extent.

# Kinetic energy (1)

**Short:** KS = Accurate kinetic energy by re-introducing orbitals.

Exact form of the kinetic energy:

$$T = \sum_i n_i \langle \omega_i | -\frac{1}{2} \nabla^2 | \omega_i \rangle$$

and the density:

$$n(\mathbf{r}) = \sum_i n_i \sum_s |\omega_i(\mathbf{r}, s)|^2$$

where  $n_i$  are fractional occupation numbers ( $0 \leq n_i \leq 1$ ) and  $\omega_i$  are natural orbitals. These can be derived from a general antisymmetric N-body wavefunction. (In case of a single Slater determinant,  $n_i$  is integer.)

89/99

# Kinetic energy (2)

Kohn and Sham introduce a set of auxiliary orbitals,  $\psi_i(\mathbf{r}, s)$ , as new degrees of freedom: *the non-interacting reference system*. The density is derived from the orbitals as:

$$n(\mathbf{r}) = \sum_{i=1}^N n_i \sum_s |\psi_i(\mathbf{r}, s)|^2.$$

$n(\mathbf{r})$  can be used to evaluate all parts of the energy functional, except the kinetic energy.

It can be shown that for any density derived from a general antisymmetric wavefunction, there is a corresponding *non-interacting reference system*.

## Kinetic energy (3)

The kinetic energy is *approximated* as:

$$T[n] \approx T_s[n] = \sum_{i=1}^N n_i \langle \psi_i | -\frac{1}{2} \nabla^2 | \psi_i \rangle$$

In many implementations,  $n_i$ , is limited to 0 or 1, but in principle fractional values are allowed. It is only exact if  $\psi_i$  are the natural orbitals.

The universal functional takes the form

$$F_{\text{KS}}[n] = T_s[n] + J[n] + E_{\times c}[n]$$

where the last term is the exchange-correlation functional. It consists of two *corrections*:

$$E_{xc}[n] = (T[n] - T_s[n]) + (V_{ee}[n] - J[n])$$

91/99

## The Kohn-Sham equations

KS & HF ground state are computed with the same mathematical tools.

$$\frac{\delta E_{KS}}{\delta n_i \psi_i^*(\mathbf{x})} = \epsilon_i \psi_j(\mathbf{x})$$

with

$$\frac{\delta E_{KS}}{\delta n_i \psi_i^*(\mathbf{x})} = -\frac{1}{2} \nabla^2 \psi_i(\mathbf{x}) + V_{\text{eff}}[n](\mathbf{r}) \psi_i(\mathbf{x})$$

and

$$V_{\text{eff}}[n](\mathbf{r}) = V_{\text{ext}}(\mathbf{r}) + \frac{\delta J[n]}{\delta n(\mathbf{r})} + \frac{\delta E_{\text{xc}}[n]}{\delta n(\mathbf{r})}$$

$$= V_{\text{ext}}(\mathbf{r}) + V_{\text{H}}[n](\mathbf{r}) + V_{\text{xc}}[n](\mathbf{r})$$

The ground state KS-orbitals (except HOMO) have no physical interpretation!

# The Kohn-Sham equations in a finite basis set

The same approach as in Hartree-Fock, i.e. construct a *Fock* matrix.

$$[F]_{km} = \frac{\partial E_{\text{KS}}}{\partial [D]_{mk}}$$

The Kohn-Sham equations in matrix form take a similar form as the Rothean equations:

$$FC = SC\epsilon$$

The exchange-correlation contribution,  $\frac{\partial E_{xc}}{\partial [D]_{mk}}$ , requires numerical integration techniques!

93/99

Aufbau and Hund rules are not strictly valid in DFT

When **integer populations** are enforced:

- *Filling up the  $N$  lowest orbitals does not always lead to the ground state.*
  - In many cases the rules are still valid. Hence, the simple SCF procedure often works.
  - Problematic cases include heavily distorted molecular geometries.

When **fractional populations** are allowed:

- More advanced SCF variants are needed to optimize the fractional occupations.
  - The lowest orbitals are occupied.
  - Only when the HOMO is degenerate, fractional occupations may occur at this energy level.

# The Kohn-Sham energy

One can not simply add up the orbital energies to obtain the Kohn-Shan energy. The following relations are valid:

$$\begin{aligned} E_{\text{KS}} &= T_s[n] + J[n] + E_{\text{xc}}[n] + V_{\text{ne}}[n(\mathbf{r})] \\ &= \sum_i \epsilon_i - J[n] + E_{\text{xc}}[n] - \int V_{\text{xc}}[n](\mathbf{r}) n(\mathbf{r}) d\mathbf{r} \end{aligned}$$

Interpretation of  $\epsilon_i$  is only possible for the HOMO: Janak's Theorem (Koopman's Theorem for DFT)

$$\epsilon_N \approx -I_N$$

where  $I_N$  is the ionization potential of the **Highest Occupied Molecular Orbital**.

95/99

## Spin Density Functional Theory (1)

Three Kohn-Sham schemes are widely used:

## Restricted closed shell KS: $N$ is even (RKS)

$$\Psi = \det[\alpha\phi_1 \dots \alpha\phi_{N/2} \beta\phi_1 \dots \beta\phi_{N/2}](\mathbf{x}_1, \dots, \mathbf{x}_N)$$

## Restricted open shell KS: $N_\alpha > N_\beta$ (ROKS)

$$\Psi = \det[\alpha\phi_1 \dots \alpha\phi_{N_\alpha} \beta\phi_1 \dots \beta\phi_{N_\beta}](\mathbf{x}_1, \dots, \mathbf{x}_N)$$

### Unrestricted KS: $N_\alpha > N_\beta$ (UKS)

$$\Psi = \det[\alpha\phi_1^\alpha \dots \alpha\phi_{N_\alpha}^\alpha \beta\phi_1^\beta \dots \beta\phi_{N_\beta}^\beta](\mathbf{x}_1, \dots, \mathbf{x}_N)$$

## Spin Density Functional Theory (2)

In spin-polarized DFT, the functional depends on the  $\alpha$  and  $\beta$  densities.

- Allows for coupling with an external magnetic field  $B$ .
  - Crucial for open-shell systems (without  $B$ ).

The functional becomes:

$$F[n_\alpha, n_\beta] = T_s[n_\alpha, n_\beta] + J[n_\alpha + n_\beta] + E_{xc}[n_\alpha, n_\beta]$$

## Spin Density Functional Theory (3)

For open and closed shell-systems:

- $T_s[n_\alpha, n_\beta] = T_s[n_\alpha, 0] + T_s[0, n_\beta] = T_s[n_\alpha] + T_s[n_\beta]$ .
  - $E_x[n_\alpha, n_\beta] = E_x[n_\alpha, 0] + E_x[0, n_\beta] = E_x[n_\alpha] + E_x[n_\beta]$ .
  - $E_c[n_\alpha, n_\beta] \neq E_c[n_\alpha, 0] + E_c[0, n_\beta]$ .

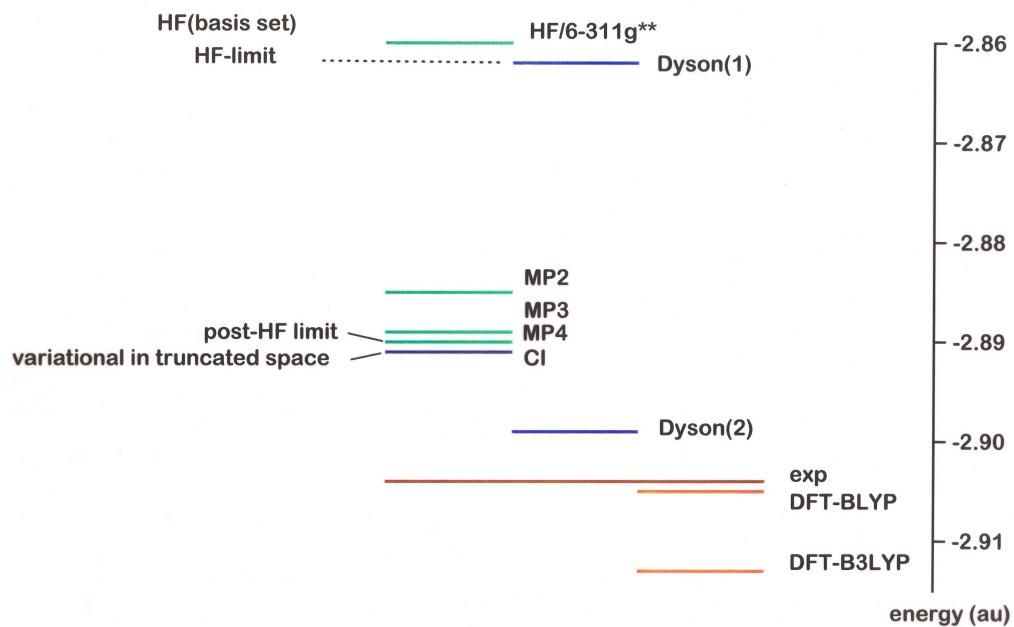
For closed shell-systems:

- $F[n] = F[\frac{1}{2}n, \frac{1}{2}n]$ .
  - $T_s[n] = T_s[\frac{1}{2}n, \frac{1}{2}n] = T_s[\frac{1}{2}n, 0] + T_s[0, \frac{1}{2}n] = 2T_s[\frac{1}{2}n, 0]$
  - $E_x[n] = E_x[\frac{1}{2}n, \frac{1}{2}n] = E_x[\frac{1}{2}n, 0] + E_x[0, \frac{1}{2}n] = 2E_x[\frac{1}{2}n, 0]$

These rules relate the kinetic energy and the exchange terms in a HK-DFT functional to the corresponding terms in a KS-SDFT functional.

# Spin Density Functional Theory (4)

## Binding energy in He



# Computational Physics: Density Functional Theory (DFT) Part 2

Toon Verstraelen

[toon.verstraelen@ugent.be](mailto:toon.verstraelen@ugent.be)

Ghent University



Academic Year 2012-2013, First Semester

## Outline

- 1 Atomic DFT
- 2 Orbital basis sets
- 3 Gaussian integrals
- 4 Functionals
- 5 Numerical integrations
- 6 DFT Limitations

# Organization

Last week

- Mostly theoretical foundations.
- Hartree-Fock (HF), Density matrix, HF limitations, Density Functional Theory (DFT).

This week

- Mostly computational aspects of HF and DFT.
- Technical background needed for practical sessions.
- Some example applications.

A large number of commercial or open-source DFT codes is available. A tentative list can be found here:

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



Atomic DFT computations

## Why atomic DFT computations?

- Useful physics in general.
  - Provides many insights required for molecular computations.
  - The computational methods are similar to molecular computations.
  - Parts of the solution can be obtained analytically.

5 / 73

## Kohn-Sham DFT basics

KS-DFT gives approximate solutions for the many-body electron problem by solving an *auxiliary* one-body Schrödinger equation.

$$\left[ -\frac{1}{2} \nabla^2 + V[n]_{\text{eff}}(\mathbf{r}) \right] \psi_i = \epsilon_i \psi_i$$

with:

- $V_{\text{eff}}[n]$ : the effective potential.
  - $\psi_i$ : the orbitals in the Slater determinant.
  - $n(\mathbf{r}) = \sum_{i=1}^N |\psi_i(\mathbf{r})|^2$ : the electron density.

Due to the dependence of  $V$  on  $n$ , and hence on  $\psi$ , the equations must be solved iteratively:  $n^0(\mathbf{r}) \Rightarrow V_{\text{eff}}[n^0](\mathbf{r}) \Rightarrow n^1(\mathbf{r}) \Rightarrow V_{\text{eff}}[n^1](\mathbf{r}) \Rightarrow \dots$

## Central potential

Atomic  $V_{\text{eff}}$  can be made central

**Problem:** For a given **central**  $V_{\text{eff}}$ ,  $N_\alpha$  and  $N_\beta$ , one obtains in general a  $\Psi$  and  $n(\mathbf{r})$  that are not spherically symmetric. Hence  $V_{\text{eff}}$  in the next iteration is no longer central.

**Approximation:** rotationally average the density in each iteration,

$$\langle n(r) \rangle_{\text{rot}} = \frac{1}{4\pi} \int d\Omega n(\mathbf{r})$$

and derive the effective potential from  $\langle n(r) \rangle_{\text{rot}}$ .

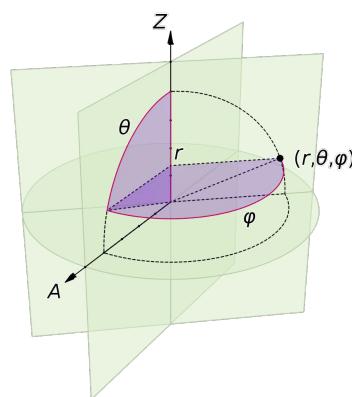
For some atoms (H, He, Li, Be, Ne, ...) this is not an approximation.

7 / 73

## Central potential

## Separation into radial and angular problems (1)

## Spherical coordinates



## The Laplacian operator in spherical coordinates:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \varphi} \frac{\partial}{\partial \varphi} \left( \sin \varphi \frac{\partial f}{\partial \varphi} \right) + \frac{1}{r^2 \sin^2 \varphi} \frac{\partial^2 f}{\partial \theta^2}$$

8 / 73

## Central potential

## Separation into radial and angular problems (2)

In spherical coordinates, the Kohn-Sham equations can be separated into radial and angular terms:

$$\begin{aligned} & \left[ -\frac{1}{2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial}{\partial r} \right) + r^2 V_{\text{eff}}(r) - r^2 \epsilon_i \right] \psi_i(\mathbf{r}) \\ &= \left[ \frac{1}{2 \sin \varphi} \frac{\partial}{\partial \varphi} \left( \sin \varphi \frac{\partial}{\partial \varphi} \right) + \frac{1}{2 \sin^2 \varphi} \frac{\partial^2}{\partial \theta^2} \right] \psi_i(\mathbf{r}) \end{aligned}$$

which can be written more compactly:

$$\hat{R}\psi_i(\mathbf{r}) = \hat{W}\psi_i(\mathbf{r})$$

## Central potential

## Separation into radial and angular problems (3)

We try the following ansatz for the solution (separation of variables):

$$\psi_i(\mathbf{r}) = W_k(\varphi, \theta) R_n(r)$$

with  $i = (k, n)$  and rewrite the KS equations as

$$\frac{\hat{R}R_n(r)}{R_n(r)} = \frac{\hat{W}W_k(\varphi, \theta)}{W_k(\varphi, \theta)}$$

The left and right hand side depend on different variables. A solution is only possible if both sides are constant:

$$\frac{\hat{R}R_n(r)}{R_n(r)} = \frac{\hat{W}W_k(\varphi, \theta)}{W_k(\varphi, \theta)} = L$$

## The angular problem

The angular problem is a standard equation in physics

$$\left[ \frac{1}{2 \sin \varphi} \frac{\partial}{\partial \varphi} \left( \sin \varphi \frac{\partial}{\partial \varphi} \right) + \frac{1}{2 \sin^2 \varphi} \frac{\partial^2}{\partial \theta^2} \right] W_k(\varphi, \theta) = L W_k(\varphi, \theta)$$

The solutions are the spherical harmonics:

$$W_k(\varphi, \theta) = Y_m^\ell(\varphi, \theta)$$

with  $k = (\ell, m)$  and

$$L = -\frac{\ell(\ell+1)}{2} \quad \text{with} \quad \ell = 0, 1, 2, \dots$$

11 / 73

## The radial problem (1)

The radial Kohn-Sham equation takes the form (after dividing by  $r$ )

$$\left[ -\frac{1}{2r} \frac{d}{dr} \left( r^2 \frac{d}{dr} \right) + rV_{\text{eff}}(r) - r\epsilon_{n\ell} \right] R_{n\ell}(r) = -\frac{\ell(\ell+1)}{2r} R_{n\ell}(r)$$

This can be rewritten as a conventional Schrödinger equation with the substitution  $R_{nl} = \frac{U_{nl}}{r}$ . The substitution in the first term leads to

$$-\frac{1}{2r} \frac{d}{dr} \left( r^2 \frac{d}{dr} \frac{U_{n\ell}}{r} \right) = -\frac{1}{2r} \frac{d}{dr} \left( r \frac{dU_{n\ell}}{dr} - U_{n\ell} \right) = -\frac{1}{2} \frac{d^2 U_{n\ell}}{dr^2}$$

## Central potential

## The radial problem (2)

Substitution of  $R_{nl} = \frac{U_{nl}}{r}$  in the remainder of the radial KS-equation yields finally:

$$\left[ -\frac{1}{2} \frac{d^2}{dr^2} + \frac{\ell(\ell+1)}{2r^2} + V_{\text{eff}}(r) \right] U_{n\ell}(r) = \epsilon_{n\ell} U_{n\ell}(r)$$

- first term: radial kinetic energy
  - second term: angular kinetic energy (centrifugal force)
  - third term: potential energy
  - right hand: orbital energy (Lagrange multiplier!)

13 / 73

## Central potential

# Combining radial and angular solutions

The most general form of an orbital is:

$$\psi(\mathbf{r}) = \sum_m A_{n\ell m} Y_m^\ell(\varphi, \theta) \frac{U_{n\ell}(r)}{r}$$

with a well-defined corresponding orbital energy,  $\epsilon_{nl}$ .

## Central potential

## Real solutions

When the potential  $V_{\text{eff}}$  is real, the Hamiltonian operator is real (and symmetric) and all orbitals can be written as real functions.

One uses real spherical harmonics (e.g. for visualization):

$$s = Y_0^0 = \frac{1}{2} \sqrt{\frac{1}{\pi}}$$

$$p_x = \sqrt{\frac{1}{2}} \left( Y_{-1}^1 - Y_1^1 \right) = \sqrt{\frac{3}{4\pi}} \cdot \frac{x}{r}$$

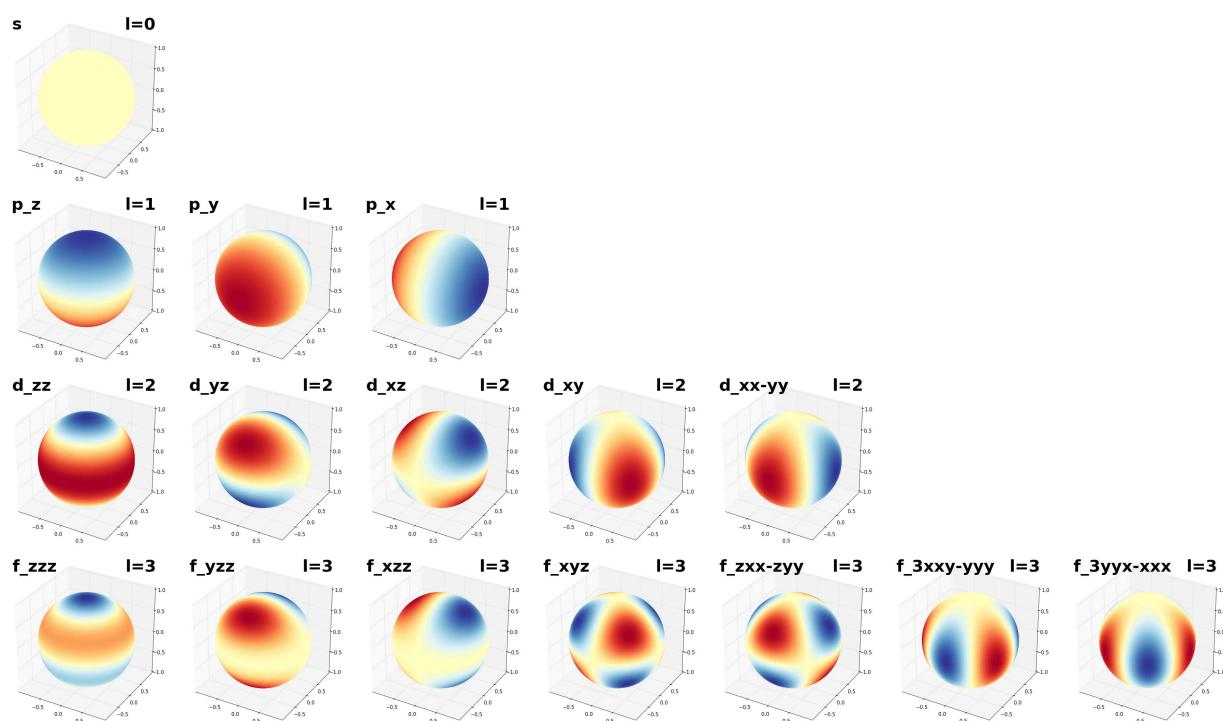
$$p_y = i\sqrt{\frac{1}{2}} \left( Y_{-1}^1 + Y_1^1 \right) = \sqrt{\frac{3}{4\pi}} \cdot \frac{y}{r}$$

$$p_z = Y_1^0 = \sqrt{\frac{3}{4\pi}} \cdot \frac{z}{r}$$

• • •

## Central potential

## Terminology: real spherical harmonics



# Hydrogen-like atom (1)

For hydrogen-like atoms, the potential is known exactly:

$V_{\text{eff}} = V_{\text{ext}} = -Z/r$ . The radial equation has known analytical solutions. (Laguerre polynomials) The orbital energy does not depend on the angular quantum number,  $\ell$ :

$$\epsilon_{n\ell} = -\frac{Z^2}{2n^2} \text{ where } n = \ell + 1, \ell + 2, \dots$$

These are the first few orbitals.

$$\psi_{n=1,s} = Z^{\frac{3}{2}} \frac{1}{\sqrt{\pi}} \exp(-Zr)$$

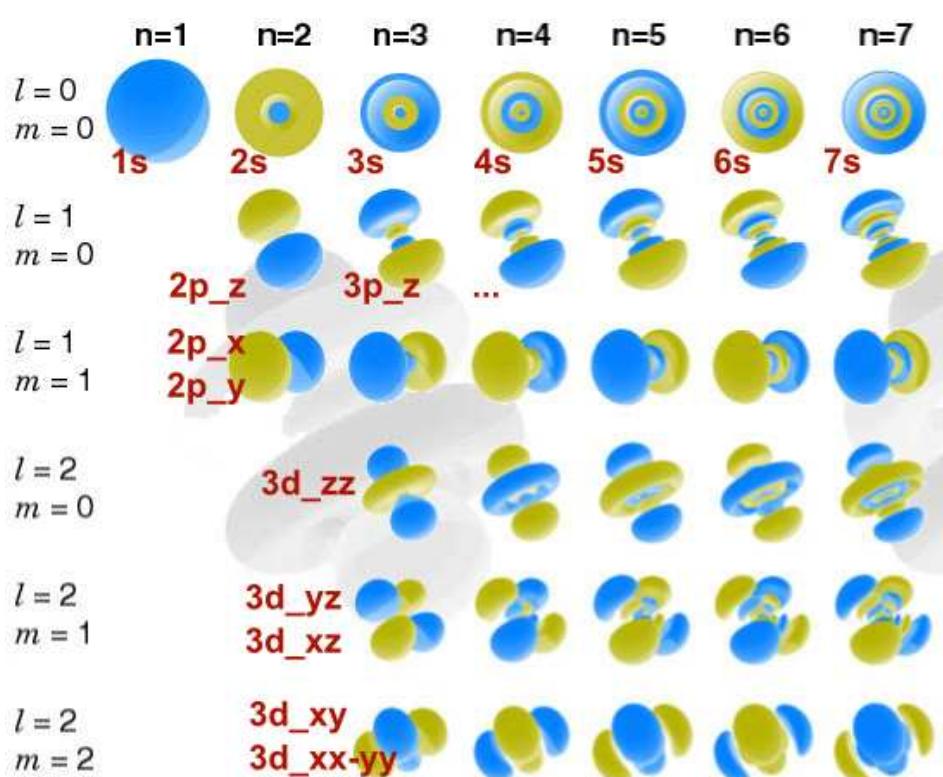
$$\psi_{n=2,s} = Z^{\frac{3}{2}} \frac{1}{4\sqrt{2\pi}} \exp(-Zr/2)(2 - Zr)$$

$$\psi_{n=2,p_x} = Z^{\frac{3}{2}} \frac{1}{4\sqrt{2\pi}} \exp(-Zr/2)x$$

$$\psi_{n=3,s} = Z^{\frac{3}{2}} \frac{1}{81\sqrt{3\pi}} \exp(-Zr/3)(27 - 18Zr + 2(Zr)^2)$$

...

# Hydrogen-like atom (2)



## Examples

# The periodic table

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1 <b>H</b> Hydrogen 1.00794	2 <b>He</b> Helium 4.00202	3 <b>Li</b> Lithium 6.941	4 <b>Be</b> Boron 9.01216	5 <b>Na</b> Boron 22.9897029	6 <b>Mg</b> Magnesium 24.3050	7 <b>Al</b> Aluminum 26.9813886	8 <b>Si</b> Silicon 28.0855	9 <b>P</b> Phosphorus 30.973762	10 <b>S</b> Sulfur 32.065	11 <b>Cl</b> Chlorine 35.453	12 <b>Ar</b> Argon 39.948	13 <b>Br</b> Bromine 79.904	14 <b>Kr</b> Krypton 83.798	15 <b>Ca</b> Calcium 40.078	16 <b>Sc</b> Scandium 44.95912	17 <b>Sc</b> Scandium 44.95912	18 <b>He</b> Helium 4.00202
<b>1</b> Atomic # <b>Symbol</b> Name Atomic Mass	<b>2</b> Hydrogen 1.00794	<b>3</b> Lithium 6.941	<b>4</b> Boron 9.01216	<b>5</b> Boron 22.9897029	<b>6</b> Magnesium 24.3050	<b>7</b> Aluminum 26.9813886	<b>8</b> Silicon 28.0855	<b>9</b> Phosphorus 30.973762	<b>10</b> Sulfur 32.065	<b>11</b> Chlorine 35.453	<b>12</b> Argon 39.948	<b>13</b> Bromine 79.904	<b>14</b> Krypton 83.798	<b>15</b> Calcium 40.078	<b>16</b> Scandium 44.95912	<b>17</b> Helium 4.00202	
<b>C</b> Solid	<b>Hg</b> Liquid	<b>H</b> Gas	<b>RF</b> Unknown	<b>Metals</b>	<b>Nonmetals</b>												
				<b>Alkal metals</b>	<b>Alkaline earth metals</b>	<b>Lanthanoids</b>	<b>Actinoids</b>	<b>Transition metals</b>	<b>Poor metals</b>	<b>Other nonmetals</b>	<b>Moblie gases</b>						
19 <b>K</b> Potassium 39.093	20 <b>Ca</b> Calcium 40.078	21 <b>Sc</b> Scandium 44.95912	22 <b>Ti</b> Titanium 47.867	23 <b>V</b> Vanadium 50.9416	24 <b>Cr</b> Chromium 51.9961	25 <b>Mn</b> Manganese 54.938045	26 <b>Fe</b> Iron 55.845	27 <b>Co</b> Cobalt 58.93195	28 <b>Ni</b> Nickel 58.6934	29 <b>Cu</b> Copper 63.546	30 <b>Zn</b> Zinc 65.38	31 <b>Ga</b> Gallium 69.729	32 <b>Ge</b> Germanium 72.94	33 <b>As</b> Arsenic 74.959	34 <b>Se</b> Selenium 78.94	35 <b>Br</b> Bromine 79.904	36 <b>Kr</b> Krypton 83.798
37 <b>Rb</b> Rubidium 85.4978	38 <b>Sr</b> Strontium 87.62	39 <b>Y</b> Yttrium 87.90585	40 <b>Zr</b> Zirconium 91.224	41 <b>Nb</b> Niobium 92.90658	42 <b>Mo</b> Molybdenum 95.96	43 <b>Tc</b> Technetium 97.9072	44 <b>Ru</b> Ruthenium 101.07	45 <b>Rh</b> Rhodium 102.90550	46 <b>Pd</b> Palladium 106.42	47 <b>Ag</b> Silver 107.868	48 <b>Cd</b> Cadmium 112.411	49 <b>In</b> Indium 114.818	50 <b>Sn</b> Tin 118.710	51 <b>Sb</b> Antimony 121.760	52 <b>Te</b> Tellurium 127.69	53 <b>I</b> Iodine 126.90447	54 <b>Xe</b> Xeons 131.293
55 <b>Cs</b> Cesium 132.904519	56 <b>Ba</b> Barium 137.327	57-71 <b>89-103</b>	72 <b>Hf</b> Hafnium 178.49	73 <b>Ta</b> Tantalum 180.94788	74 <b>W</b> Tungsten 183.84	75 <b>Re</b> Rhenium 186.207	76 <b>Os</b> Osmium 190.23	77 <b>Ir</b> Iridium 192.217	78 <b>Pt</b> Platinum 195.084	79 <b>Au</b> Gold 196.96669	80 <b>Hg</b> Mercury 200.59	81 <b>Tl</b> Thallium 204.98383	82 <b>Pb</b> Lead 207.2	83 <b>Bi</b> Bismuth 208.9640	84 <b>Po</b> Polonium 208.9624	85 <b>At</b> Astatine 209.8677	86 <b>Rn</b> Radium 222.0176
87 <b>Fr</b> Francium (223)	88 <b>Ra</b> Radium (226)	104 <b>Rf</b> Rutherfordium (281)	105 <b>Db</b> Dubnium (282)	106 <b>Sg</b> Seaborgium (286)	107 <b>Bh</b> Bohrium (284)	108 <b>Hs</b> Hassium (277)	109 <b>Mt</b> Meitnerium (271)	110 <b>Ds</b> Darmstadtium (272)	111 <b>Rg</b> Roentgenium (285)	112 <b>Uub</b> Ununbium (286)	113 <b>Uut</b> Ununtrium (284)	114 <b>Uup</b> Ununpentium (286)	115 <b>Uus</b> Ununseptium (286)	116 <b>Uuo</b> Ununoctium (284)	117 <b>Uuu</b> Ununtrium (294)	118 <b>Uuo</b> Ununoctium (294)	

For elements with no stable isotopes, the mass number of the isotope with the longest half-life is in parentheses.

Design and Interface Copyright © 1997 Michael Dayah (michael@dayah.com). <http://www.ptable.com/>



19 / 73

## Examples

## Atomic electron density (1)

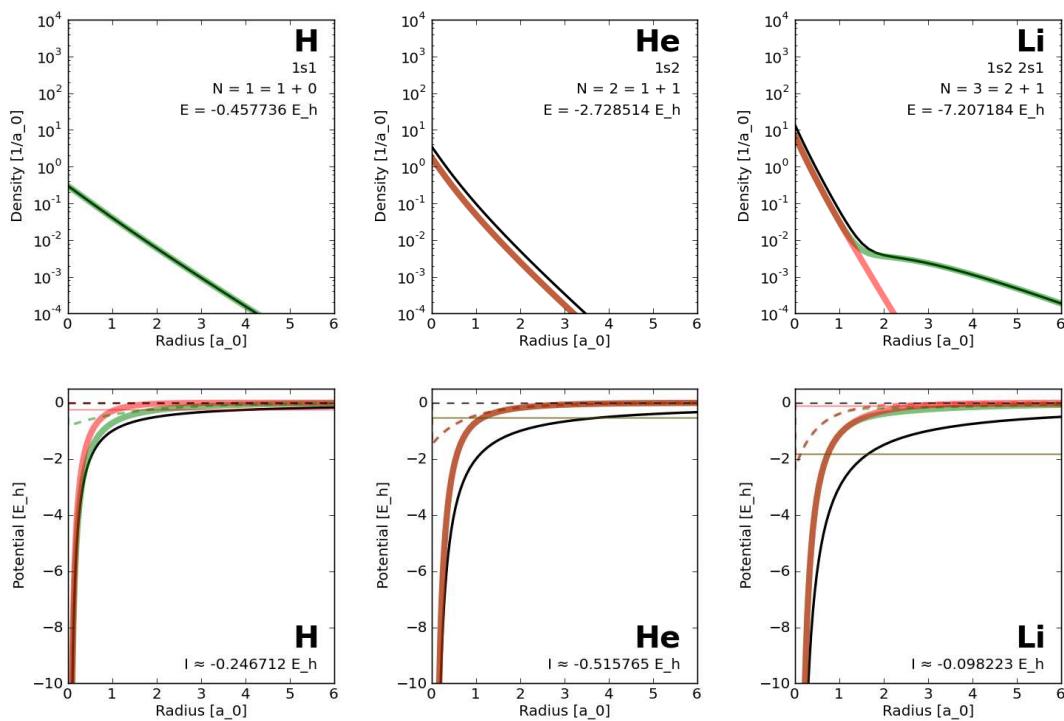
Typical atomic electron density features include

- The cusp (spike at nucleus):  $n(\mathbf{r}) \approx \exp(-2Zr)$
  - The tail:  $n(\mathbf{r}) \approx \exp(-2r\sqrt{2I})$  ( $I$  is ionization potential)
  - Approx. piecewise exponential function
  - Noble gas cores are easily recognizable.
  - *Core* density = completely filled shells
  - *Valence* density (tail) = partially filled shells
  - *Vacuum*  $\approx$  density below  $10^{-4}$  a.u.

All atomic results obtained with the HFS functional. (HFS = Only Dirac exchange, no correlation)

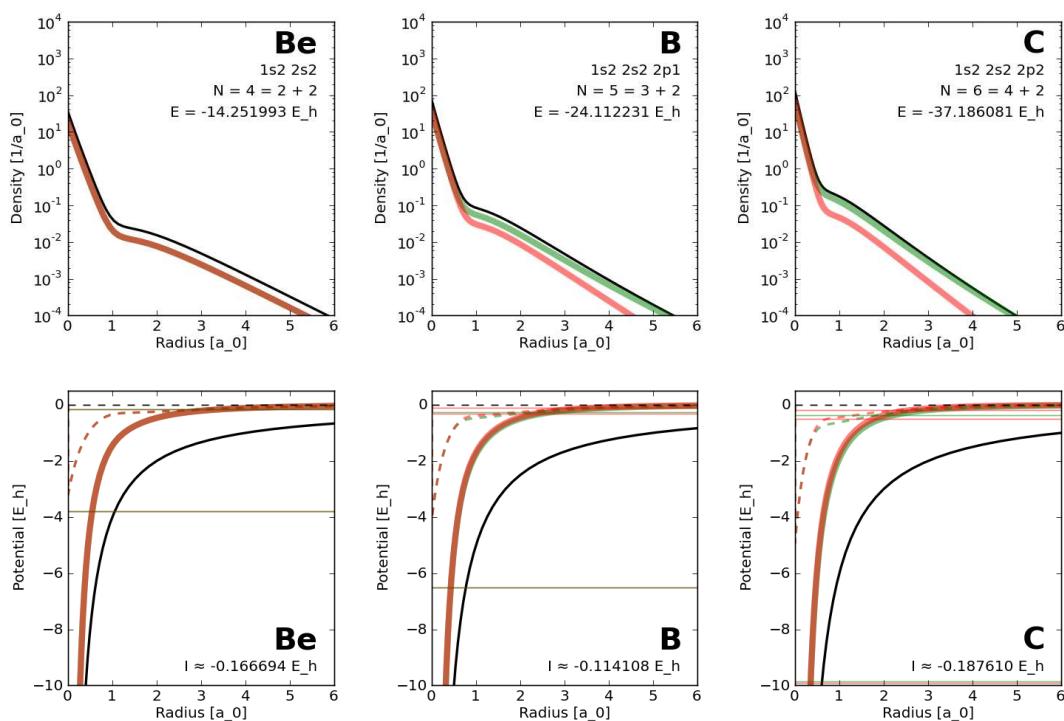
## Examples

## Atomic electron density (2)



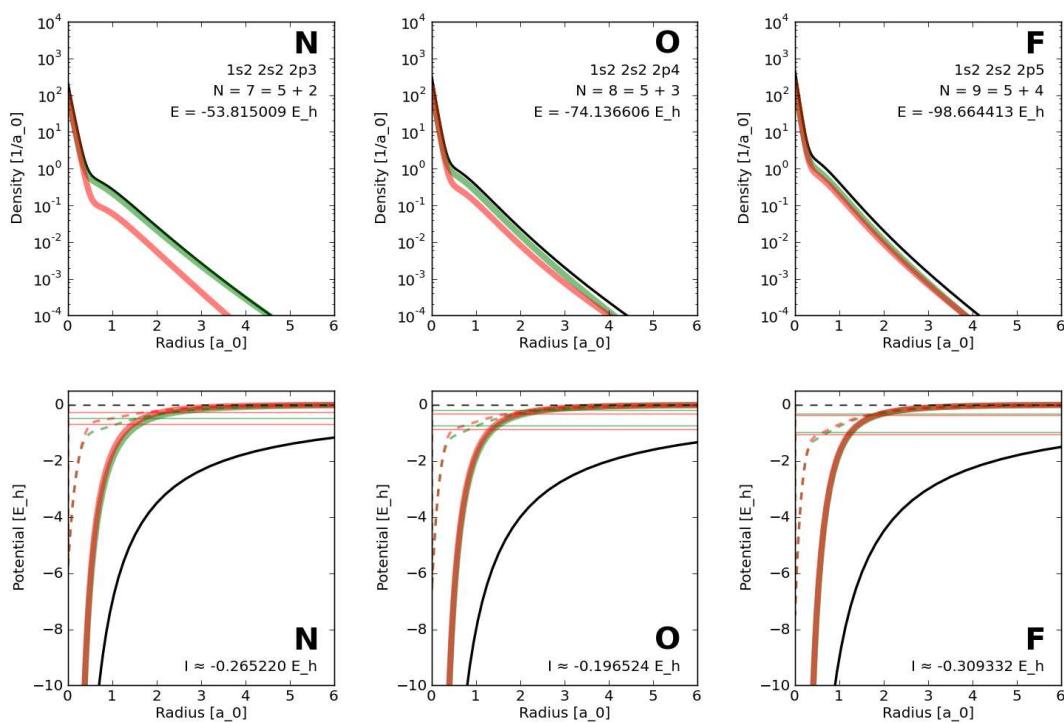
## Examples

## Atomic electron density (3)



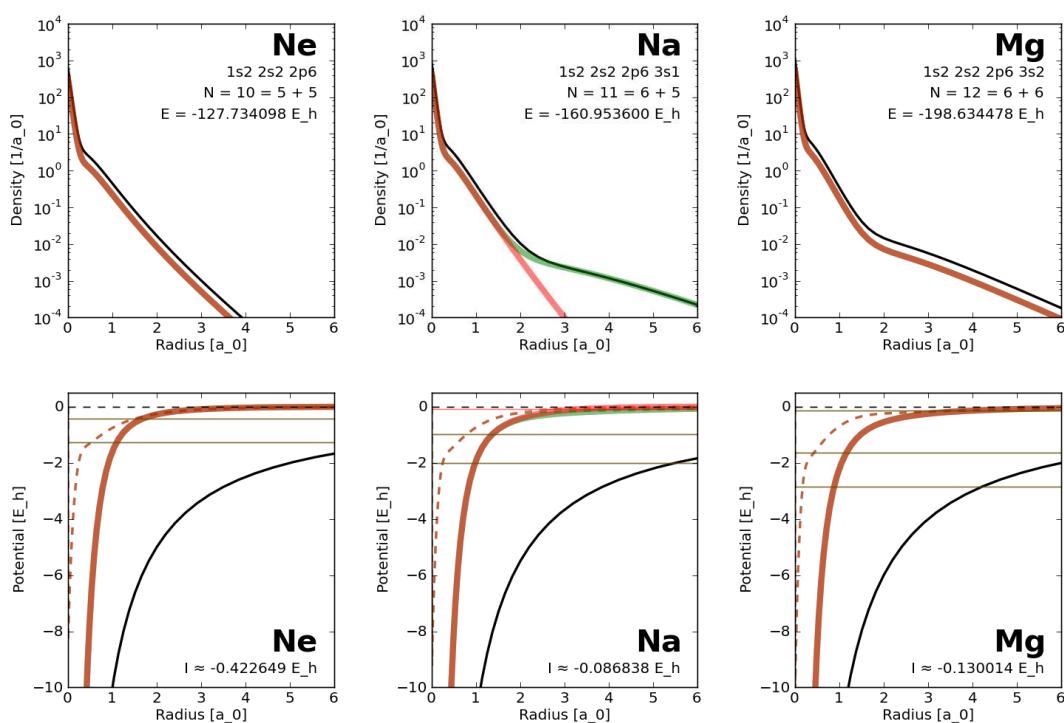
## Examples

## Atomic electron density (4)



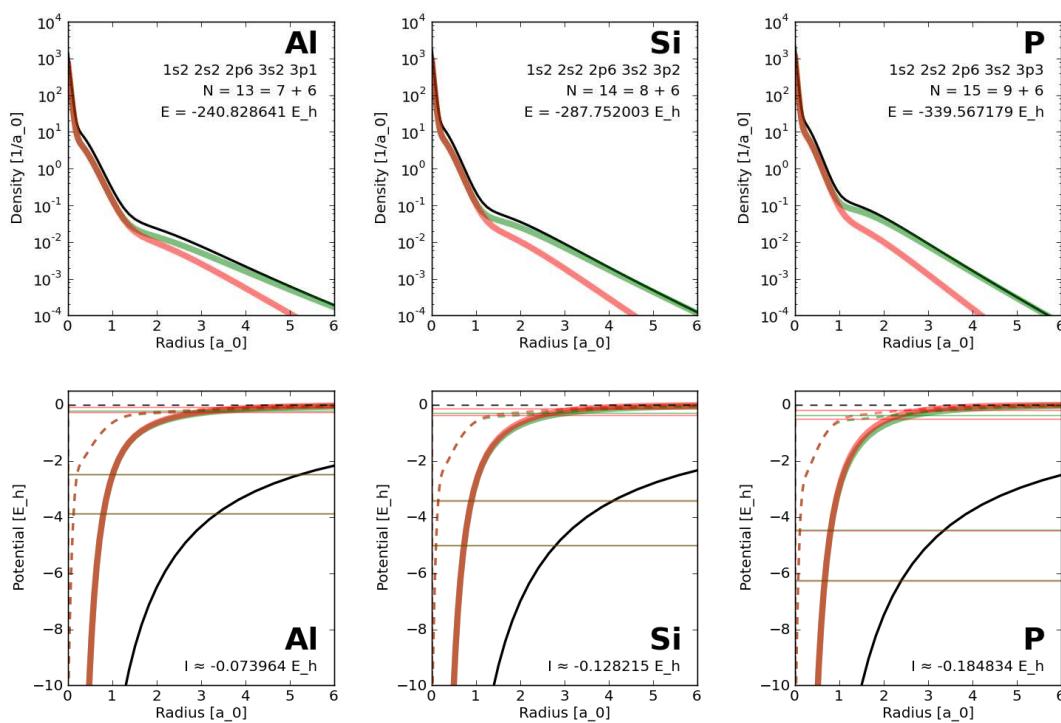
## Examples

## Atomic electron density (5)



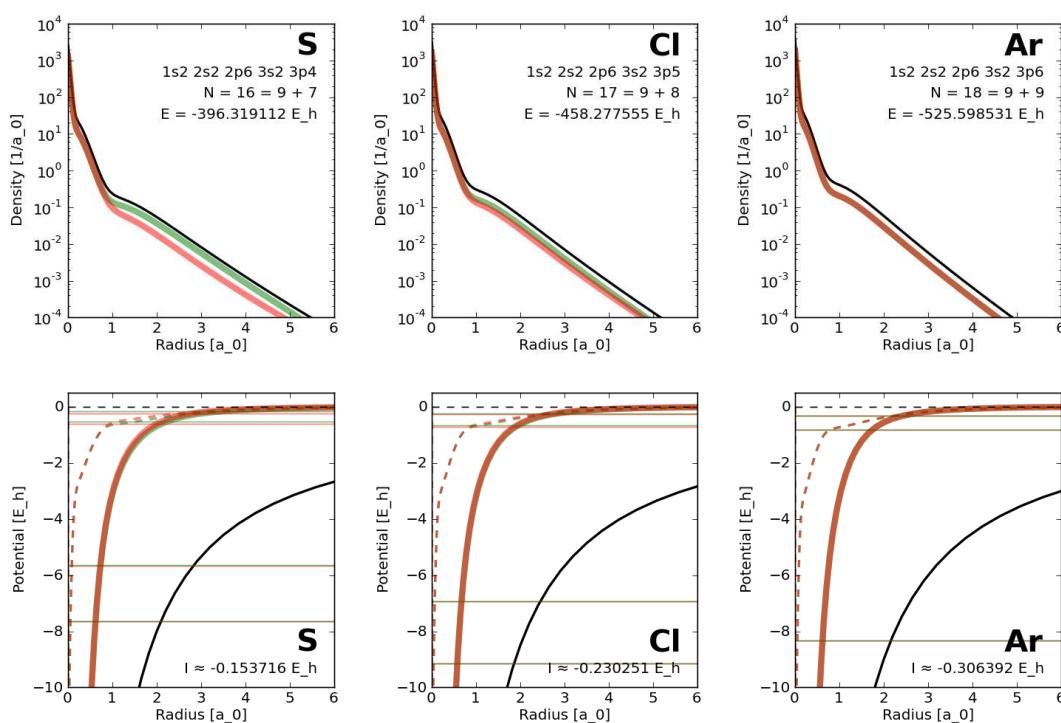
## Examples

## Atomic electron density (6)

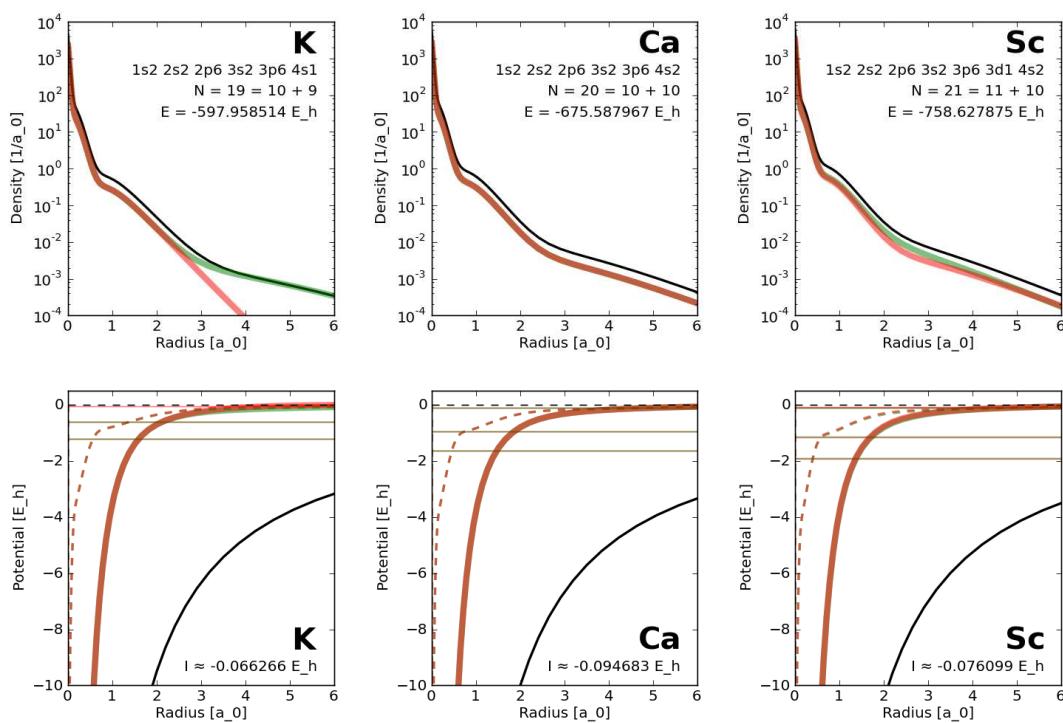


## Examples

## Atomic electron density (7)

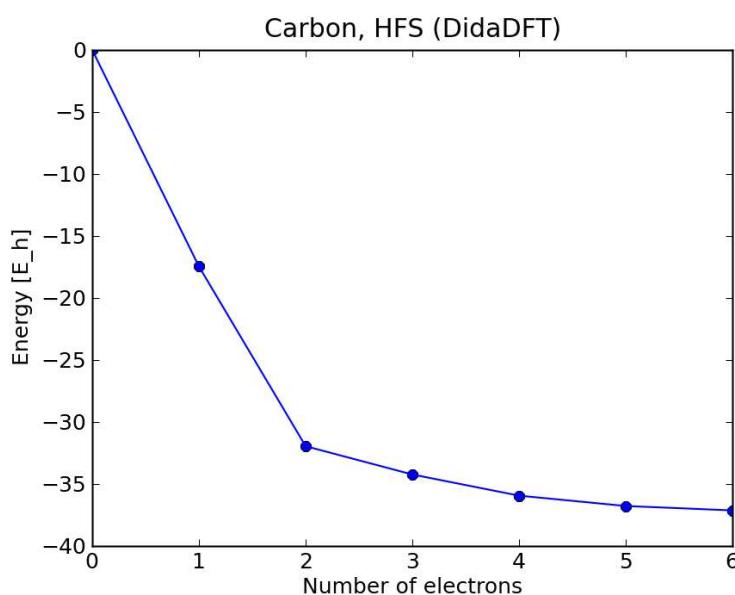


# Atomic electron density (8)



27 / 73

# The ionization energy is never negative



The picture is similar for molecules. In computational applications, one may observe negative ionization potentials due to the limitations of the basis set.

28 / 73

## Orbital basis sets

## Introduction

## Different types of orbital basis sets

- Local basis sets (linear combination of atomic orbitals, LCAO)
    - Slater type orbitals (STO)
    - **Gaussian type orbitals (GTO)**
    - Numerical atomic orbitals
  - Non-local basis sets
    - **Plane-wave basis sets**
    - Cartesian grid schemes
    - Wavelet basis sets
  - Hybrid schemes
    - Augmented Planar Waves (APW)
    - Projector Augmented Waves (PAW)

P.S. Some codes also use auxiliary basis sets for densities and effective potentials.

# Basics

It is widely accepted that atoms in molecules resemble atoms in vacuum.

The general idea behind LCAO is that each *molecular orbital* is to good approximation a linear combination of atomic orbitals:

$$\psi_i(\mathbf{r}) = \sum_{k=1}^M [C]_{ki} b_k(\mathbf{r} - \mathbf{R}_{A(k)})$$

where  $b_k$  is an atomic orbital of atom  $A$ .

It is sufficient to work with real basis functions because bound eigenstates of a Hamiltonian can always be written as real functions

31 / 73

# Slater type orbitals (STO)

Because the orbitals of a hydrogen-like (and actually any) atom decay exponentially, one could approximate the atomic orbitals by *Slater functions*

$$b_k(\mathbf{r}) = x^{\ell_{xk}} y^{\ell_{yk}} z^{\ell_{zk}} N_{\text{STO}}(\alpha, \ell_{xk}, \ell_{yk}, \ell_{zk}) \exp(-\alpha_k |\mathbf{r}|)$$

The main drawback is that there are no convenient analytical expressions for the typical integrals over Slater functions. (kinetic, NAI & ERI)

32 / 73

# Gaussian type orbitals (GTO) (1)

For computational reasons, most codes expand the atomic orbitals in (just a few) Cartesian Gaussian *primitives*:

$$b_k(\mathbf{r}) = x^{\ell_{xk}} y^{\ell_{yk}} z^{\ell_{zk}} \sum_{p=1}^{P_k} d_{pk} N_{\text{GTO}}(\alpha, \ell_{xk}, \ell_{yk}, \ell_{zk}) \exp(-\alpha_{kp} |\mathbf{r}|^2)$$

- $P_k$  is called the contraction length.
- The exponents  $\ell_{xk}, \ell_{yk}, \ell_{zk}$  are just integer parameters, not quantum numbers.
- $\ell_{xk} + \ell_{yk} + \ell_{zk}$  must match the angular quantum number,  $\ell_k$ , of the atomic orbital.
- A set of cartesian functions for given  $\ell_k$  is called a Cartesian Gaussian *shell*.
- Integrals over Gaussian functions in HF (all) and DFT (all except xc) have closed expressions.
- $N_{\text{GTO}}(\alpha, \ell_{xk}, \ell_{yk}, \ell_{zk})$  is a normalization for each primitive.

33 / 73

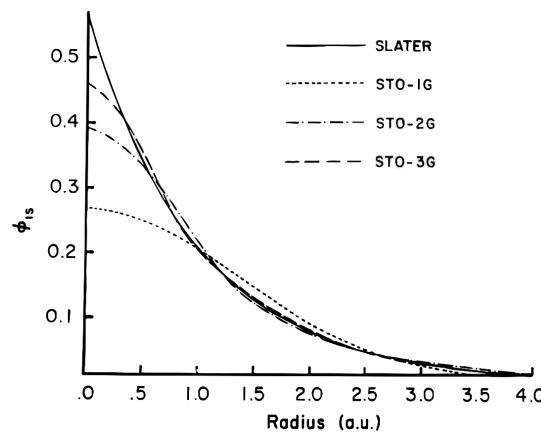
# Gaussian type orbitals (GTO) (2)

Starting from angular momentum  $\ell = 2$ , it is desirable to convert a Cartesian shell to a pure shell. In a pure shell, the angular dependence is given by the spherical harmonics. Example transformation:

$$\begin{pmatrix} X(C_2^0) \\ X(C_2^1) \\ X(S_2^1) \\ X(C_2^2) \\ X(S_2^2) \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} & 0 & 0 & -\frac{1}{2} & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{2}\sqrt{3} & 0 & 0 & -\frac{1}{2}\sqrt{3} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X(xx) \\ X(xy) \\ X(xz) \\ X(yy) \\ X(yz) \\ X(zz) \end{pmatrix}$$

## Gaussian type orbitals (GTO) (3)

**Minimal basis sets:** least squares fit to Slater functions.  
(STO-xG)



- The cusp is poorly represented (zero derivative).
- The tail decays too rapidly.

35 / 73

## Gaussian type orbitals (GTO) (4)

**Split valence:** parameters  $d_{pk}$  and  $\alpha_{pk}$  are estimated with atomic (or even molecular) HF or DFT computations.

- Core basis functions have long contractions. (Cusp)
- Split (=multiple) valence functions (double zeta, triple zeta, ...)
- Next to atomic orbitals there are two types of additional basis functions (often with  $P_k = 1$ )
  - Polarization functions: allow more deformation of (atomic) orbitals due to bonding. (local,  $\alpha_{pk}$  high)
  - Diffuse functions: improve accuracy of density tails, mostly for anionic systems. (non-local,  $\alpha_{pk}$  low)

36 / 73

# Gaussian type orbitals (GTO) (5)

## Pro:

- Small number of basis functions gives reasonable results.
- Link with atomic computations, or chemical intuition.

## Con:

- Basis functions are centered on nuclei. Different nuclear geometries have a different (molecular) basis and hence different reference energies (BSSE error).
- No systematic improvement towards complete basis.
- Not orthogonal, sometimes near-degenerate (diffuse functions).

# Gaussian type orbitals (GTO) (6)

Example (Carbon, 3-21G) taken from  
<https://bse.pnl.gov/bse/portal>

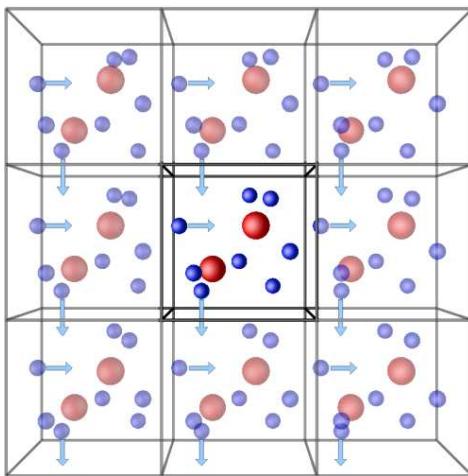
```

C      0
S      3    1.00
        172.2560000    0.0617669
        25.9109000    0.3587940
        5.5333500    0.7007130
SP     2    1.00
        3.6649800    -0.3958970    0.2364600
        0.7705450    1.2158400    0.8606190
SP     1    1.00
        0.1958570    1.0000000    1.0000000

```

# Periodic boundary conditions (1)

Modeling of crystals and liquids requires periodic boundary conditions to avoid termination effects.



A basis of plane waves automatically satisfies periodic boundaries and offer specific computational advantages.

39 / 73

# Periodic boundary conditions (2)

Given a unit cell with lattice vectors  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$ . The reciprocal lattice vectors  $\tilde{\mathbf{a}}$ ,  $\tilde{\mathbf{b}}$ ,  $\tilde{\mathbf{c}}$  are bi-orthogonal to the real lattice vectors:

$$\mathbf{a} \cdot \tilde{\mathbf{a}} = 1$$

$$\mathbf{a} \cdot \tilde{\mathbf{b}} = 0$$

$$\mathbf{a} \cdot \tilde{\mathbf{c}} = 0$$

$$\mathbf{b} \cdot \tilde{\mathbf{a}} = 0$$

$$\mathbf{b} \cdot \tilde{\mathbf{b}} = 1$$

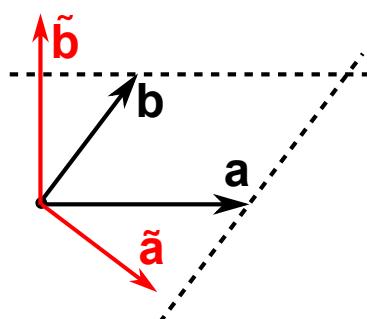
$$\mathbf{b} \cdot \tilde{\mathbf{c}} = 0$$

$$\mathbf{c} \cdot \tilde{\mathbf{a}} = 0$$

$$\mathbf{c} \cdot \tilde{\mathbf{b}} = 0$$

$$\mathbf{c} \cdot \tilde{\mathbf{c}} = 1$$

2D example:



40 / 73

## Periodic boundary conditions (3)

The general form of the plane wave basis is:

$$b_{k_a k_b k_c}(\mathbf{r}) = \exp\left(i(k_a \tilde{\mathbf{a}} + k_b \tilde{\mathbf{b}} + k_c \tilde{\mathbf{c}}) \cdot \mathbf{r}\right)$$

where the indices  $k_a$ ,  $k_b$  and  $k_c$  are integers. One may truncate the basis by defining a maximum for the kinetic energy of the plane waves.

41 / 73

## Periodic boundary conditions (4)

### Pro:

- No BSSE
- Systematic improvement towards complete basis.
- Orthogonal.
- Can represent the Hartree potential.

### Con:

- Major: many plane waves required to describe cusp
- Minor: inefficient in case of porous media (basis functions for vacuum)

42 / 73

## Intgrals over Gaussian basis functions

## Scope (1)

Many integrals can be computed analytically with Gaussian basis functions (centered at different nuclei):

- Overlap:  $[S]_{k\ell} = \int b_k^*(\mathbf{x}) b_\ell(\mathbf{x}) d\mathbf{x} \neq \delta_{k\ell}$
- Kinetic:  $[\mathcal{T}]_{k\ell} = \int b_k^*(\mathbf{x}) \left(-\frac{1}{2}\nabla^2\right) b_\ell(\mathbf{x}) d\mathbf{x}$
- NAI:  $[\mathcal{V}_{\text{ext}}]_{k\ell} = \int b_k^*(\mathbf{x}) V_{\text{ext}}(\mathbf{r}) b_\ell(\mathbf{x}) d\mathbf{x}$
- ERI:  $[\mathcal{V}_{\text{ee}}]_{k\ell m n} = \int b_k^*(\mathbf{x}_1) b_\ell^*(\mathbf{x}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} b_m^*(\mathbf{x}_1) b_n^*(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2$

Exchange and correlation integrals in DFT can only be carried out numerically.

## Scope (2)

- The derivation of general solutions will not be discussed here. (=too technical)
  - Efficient (yet complicated) recursive algorithms exist for all integrals.
  - In the practical sessions, matrices with pre-computed integrals will be provided.
  - An auxiliary note on Minerva (by Fermann and Valeev) discusses the analytical solutions. (This isn't compulsory material for the exam.)

45 / 73

## Exchange-Correlation functionals for KS-DFT

## Generations of XC functionals (Jacob's Ladder)

- ① **LDA**: Local density approximation functionals, only depend on  $n_\alpha$  and  $n_\beta$
  - ② **GGA**: Generalized gradient approximation functionals, also depend on  $|\nabla n_\alpha|^2$  and  $|\nabla n_\beta|^2$
  - ③ **Meta-GGA**: Include additional quantities  $\nabla^2 n_\alpha$ ,  $\nabla^2 n_\beta$ , kinetic energy density, ...
  - ④ **Hybrid GGA**: Include scaled Hartree-Fock exchange, or more generally any information from the occupied orbitals.
  - ⑤ **Meta-hybrid GGA**: combination of Meta-GGA and Hybrid GGA.
  - ⑥ **Double hybrid GGA**: Also include perturbation corrections (cfr. Rayleigh-Schrödinger perturbation theory), or more generally any information from the virtual orbitals.

47 / 73

## Generalized Gradient Approximation (GGA) (1)

The lowest gradient corrected (LGC) exchange functional is just based on dimensionality arguments:

$$E_x^{LGC}[n_\alpha, n_\beta] = E_x^{LDA}[n_\alpha, n_\beta] - \beta \sum_{s=\alpha, \beta} \int \frac{|\nabla n_s(\mathbf{r})|^2}{n_s^{4/3}(\mathbf{r})} d\mathbf{r}$$

$$E_x^{LDA}[n_\alpha, n_\beta] = \frac{3}{2} \left( \frac{3}{4\pi} \right)^{1/3} \sum_{s=\alpha, \beta} \int n_s^{4/3}(\mathbf{r}) d\mathbf{r}$$

- The integrand containing the gradient has the same dimension as the integrand for the LDA approximation.
  - Historically, this is the first example of gradients in exchange functionals.
  - The correction is small compared to LDA.  $\beta$  is empirical.
  - This form is numerically untractable, i.e. the corresponding potential diverges at low densities.

## Generalized Gradient Approximation (GGA) (2)

More recent GGA functions use the correcting form:

$$E_x^{GGA}[n_\alpha, n_\beta] = \frac{3}{2} \left( \frac{3}{4\pi} \right)^{1/3} \sum_{s=\alpha, \beta} \int n_s^{4/3}(\mathbf{r}) (1 + F(x_s(\mathbf{r}))) d\mathbf{r}$$

$$x_s(\mathbf{r}) = \frac{|\nabla n(\mathbf{r})|}{n^{4/3}(\mathbf{r})}$$

- $x_s(\mathbf{r})$  is the dimensionless or reduced gradient, used to construct enhancement factors ( $F$ ) to the LDA exchange functional
- Several forms for  $F$  were proposed.
- Correlation energy functionals are constructed similarly.

49 / 73

## Generalized Gradient Approximation (GGA) (3)

Some examples:

- Herman, Van Dyke, Ortenburger 1969:  $F(x) = \beta x^2$  (LGC) (no universal  $\beta$  was proposed)
- Becke 1986:  $F(x) = \beta \frac{x^2}{1+\gamma x^2}$  ( $\beta = 0.00375$ ,  $\gamma = 0.007$ )
- Becke 1988:  $F(x) = \beta \frac{x^2}{1+\frac{6\beta x}{\sinh(x)}}$  ( $\beta = 0.0042$ )
- Perdew, Burke, Ernzerhof 1996:  $F(x) = 1 + \kappa - \frac{\kappa}{1 + \frac{\mu x^2}{\kappa(24\pi^2)^{2/3}}}$  ( $\mu = 0.21951$ ,  $\kappa = 0.804$ ) This has the same form as Becke 1986, but parameters are determined differently.

With a Gaussian basis, matrix elements must be computed numerically.

50 / 73

## Generalized Gradient Approximation (GGA) (4)

**Technical details of functional development = beyond scope of this course.** The analytical forms are designed to obtain theoretically sound behavior in the limits of  $x_s$  towards zero and infinity. The remaining degrees of freedom are fixed by any combination of the following strategies:

- Functional forms (and parameters) that match accurate ab initio (post-HF) computations.
- Parameters that reproduce accurate experimental data (semi-empirical functionals).
- Parameters that satisfy theoretical limits, e.g. the Lieb-Oxford bound:  $E_x[n_\alpha, n_\beta] \geq E_{xc}[n_\alpha, n_\beta] \geq -1.679 \int n^{4/3}(\mathbf{r}) d\mathbf{r}$

Correlation functionals are designed with similar principles.

51 / 73

## Hybrid GGA functionals

**Idea:** Include a scaled (factor in range  $[0, 1]$ ) Hartree-Fock Exchange term in the XC functional.

- Originally introduced empirically
- Theoretically supported through the adiabatic connection. (Savin)
- Hybrid GGA functionals are generally more accurate in a variety of benchmarks

52 / 73

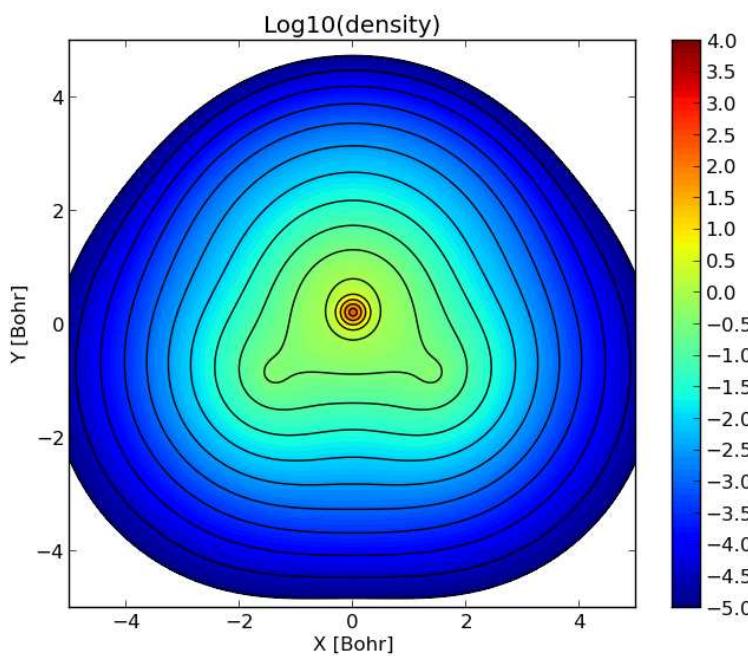
## Numerical integration techniques in DFT

## Problem (1)

- Most XC functionals lead to integrals with Gaussian functions that can not be computed analytically.
  - Simple cubic integration grids are inefficient:
    - Sharp fluctuations of the density close to the nucleus (requires fine grid).
    - Smooth variations between atoms (requires only a coarse grid).
    - No contributions far outside a molecule (requires no grid).

## Problem (2)

The  $\log_{10}$  of the  $e^-$  density of a water molecule.



55 / 73

## Overview

Conventional numerical integrations in DFT are implemented with a non-standard Gaussian quadrature technique. In the case of an LDA functional:

$$E_{\text{LDA}}[n] = \int f(n(\mathbf{r})) d\mathbf{r} \approx \sum_i^{N_p} w_i f(n(\mathbf{r}_i))$$

Specific (mostly deterministic) algorithms are used to define:

- the weights:  $w_i$
- the grid points:  $\mathbf{r}_i$

56 / 73

## Partitioning

## General partitioning concept

The integrand is first *partitioned* into atomic contributions:

$$\int f(n(\mathbf{r}))d\mathbf{r} = \sum_A^M \int w_A(\mathbf{r})f(n(\mathbf{r}))d\mathbf{r}$$

with

$$\sum_A^M w_A(\mathbf{r}) = 1 \quad 0 \leq w_A(\mathbf{r}) \leq 1$$

## Motivation:

- ① The weight functions,  $w_A(\mathbf{r})$ , are defined such that each atomic contribution has only one cusp. Other cusps are suppressed.
- ② Each atomic contribution can be carried out on a grid in spherical coordinates.

57 / 73

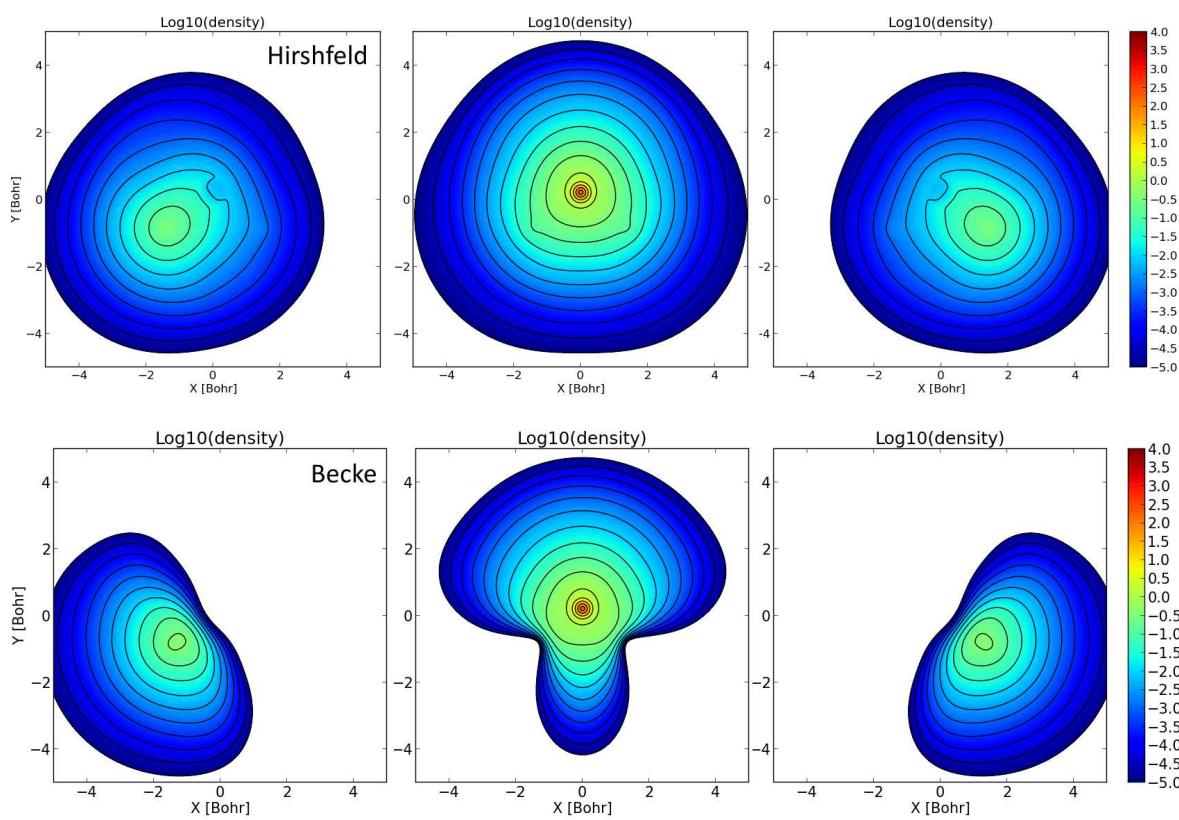
## Partitioning

## Example partitioning schemes

- Hirshfeld partitioning:  $w_A(\mathbf{r}) = \frac{n_A^0(\mathbf{r})}{\sum_B n_B^0(\mathbf{r})}$ , where  $n_B^0(\mathbf{r})$  are spherically averaged isolated-atom densities
- Becke partitioning: instead of isolated atoms, intricate polynomial switching functions are used.
  - Becke scheme is faster to evaluate  $w_A$ , but partitioned integrand is less spherical.
  - Becke was the first to propose partitioning as an integration technique.
  - Becke scheme is still widely in use.
  - Many other partitioning schemes exist, but are only used as atoms-in-molecules (AIM) analysis.

## Partitioning

## Partitioning of a single water molecule



59 / 73

## Gaussian quadrature in spherical coordinates

## Separation into radial and angular problems

The remaining difficulty is to integrate a function,  $g(r, \theta, \phi)$ , with one spike in the origin, i.e. due to the electron cusp.

The integral in spherical coordinates is separated into radial and angular parts, with specific algorithms for each part:

- ① The spherical average (for a given radius) is defined as

$$\tilde{g}(r) = \frac{1}{4\pi} \int_0^\pi d\theta \sin\theta \int_0^{2\pi} d\phi g(r, \theta, \phi)$$

- ② Then the radial part is a one-dimensional integral with the spherical average:

$$I = 4\pi \int_0^\infty r^2 \tilde{g}(r) dr$$

60 / 73

# The radial integral (1)

- Many grid points are needed close to the origin
- Only a few points are needed at distances beyond  $5a_0$  from the origin. (The density decays exponentially.)
- One typically introduces a transformation, e.g.

$$r(t) = r_{\max} \frac{\ln \left( 1 - \left( \frac{t}{N_r+1} \right)^2 \right)}{\ln \left( 1 - \left( \frac{N_r}{N_r+1} \right)^2 \right)}$$

with  $t = 1 \dots N_r$ . This grid is very fine close to the origin.

- The integral is transformed to the  $t$  variable

$$I = 4\pi \int_0^{N_r+1} r^2(t) \tilde{g}(r(t)) \frac{dr}{dt} dt$$

# The radial integral (2)

The Simpson rule for integrations on equidistant grids can be used to obtain a Gaussian quadrature formula:

$$I = 4\pi \int_0^{N_r+1} r^2(t) \tilde{g}(r(t)) \frac{dr}{dt} dt$$

$$\approx 4\pi \sum_i^{N_r} w_i^S r^2(i) \tilde{g}(r(i)) \left. \frac{dr}{dt} \right|_{t=i}$$

with

$$w_1 = w_{N_r-1} = 17/48$$

$$w_2 = w_{N_r-2} = 59/48$$

$$w_3 = w_{N_r-3} = 43/48$$

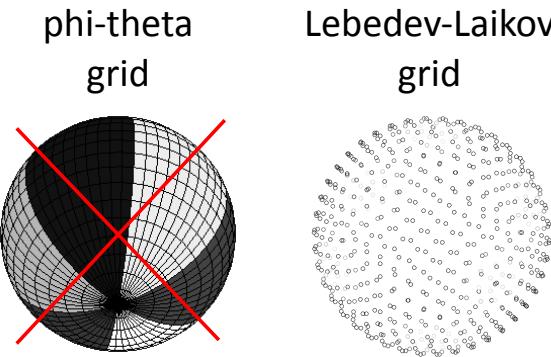
$$w_4 = w_{N_r-4} = 49/48$$

$$w_i = 1 \quad \forall i \in 5 \dots N_r - 5$$

# The angular integral

In principle, one may split the angular integral also into two one-dimensional integrals, however ...

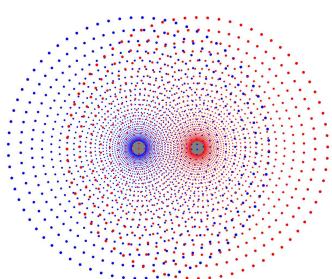
- Separate  $\phi, \theta$  grids lead to a large number of grid points at the *poles*. (=inefficient and preferred directions)
- In practice one always uses Lebedev-Laikov grids.



63 / 73

# Putting it all together

The molecular grid consists of a superposition of spherical atom-centered grids:



## The integration weights in

$$F[n] \approx \sum_i^{N_p} w_i f(n(\mathbf{r}_i))$$

are a product of five contributions:

- Partitioning weights
- Lebedev-Laikov grid weights
- Simpson weights for the radial integrals
- Transformation from  $r$  to  $t$  in radial integrals, i.e.  $\frac{dr}{dt}$
- The factor  $4\pi r^2$

64 / 73

# The functional in terms of expansion coefficients (1)

The functional becomes a function of the expansion coefficients:

$$E_{\text{LDA}}[n] = \int f \left( \sum_{k\ell}^{N_b} [D]_{kl} b_k^*(\mathbf{x}) b_\ell(\mathbf{x}) \right) d\mathbf{x}$$

The corresponding Fock matrix contribution is derived with the chain rule:

$$\begin{aligned} [\mathcal{V}_{\text{LDA}}]_{k\ell} &= \frac{\partial E_{\text{LDA}}[n]}{\partial D_{k\ell}} = \int \frac{\delta E_{\text{LDA}}[n]}{\delta n(\mathbf{x})} b_k^*(\mathbf{x}) b_\ell(\mathbf{x}) d\mathbf{x} \\ &= \int \frac{\partial f(n)}{\partial n} b_k^*(\mathbf{x}) b_\ell(\mathbf{x}) d\mathbf{x} \end{aligned}$$

# The functional in terms of expansion coefficients (2)

When the integration is carried out with a quadrature method, the same methodology is used:

$$E_{\text{LDA}}[n] \approx \sum_i^{N_p} w_i f \left( \sum_{k\ell}^{N_b} [D]_{kl} b_k^*(\mathbf{x}_i) b_\ell(\mathbf{x}_i) \right)$$

The corresponding Fock matrix contribution is derived with the chain rule:

$$\begin{aligned} [\mathcal{V}_{\text{LDA}}]_{k\ell} &\approx \frac{\partial E_{\text{LDA}}[n]}{\partial D_{k\ell}} \\ &\approx \sum_i^{N_p} w_i \frac{\partial f(n)}{\partial n} b_k^*(\mathbf{x}_i) b_\ell(\mathbf{x}_i) \end{aligned}$$

## The practical limitations of DFT

67 / 73

# Introduction

- DFT is in principle exact.
- Existing functionals are approximate.
- Compared to HF, the local correlation problem is fixed.
- Many HF issues remain and new ones are introduced.

68 / 73

# Long range behavior of the exchange potential (1)

Demonstration of failure with Dirac exchange

$$V_x \propto -n^{\frac{1}{3}} \propto -\exp(-\alpha r)$$

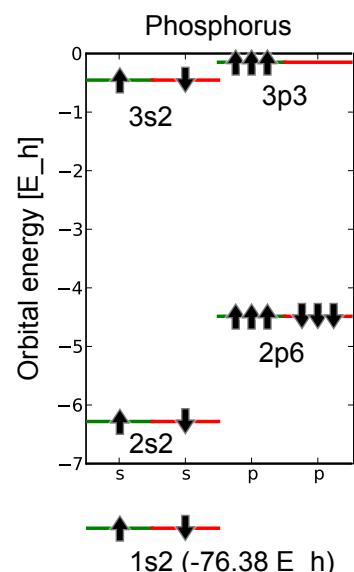
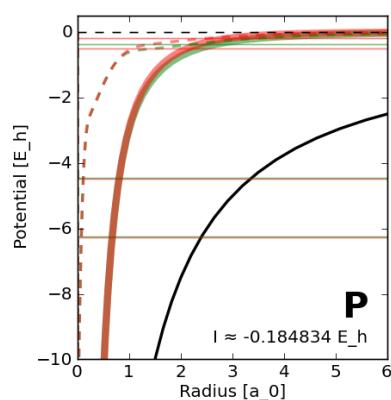
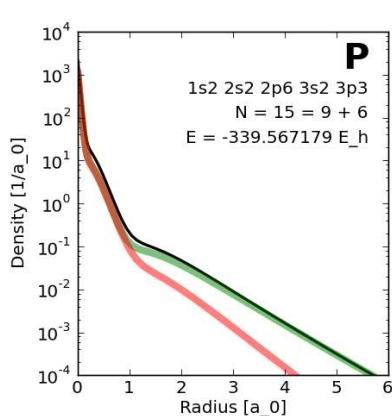
This causes unphysical behavior:

- Nuclear and Hartree potential cancel each other outside the atom.
- Exchange-Correlation remainder also decays exponentially

Using the LDA (and most GGA) approximations, a neutral atom can not bind an additional electron. **However, this is a well-known physical property of any atom or molecule (called the Rydberg series).**

69 / 73

# Long range behavior of the exchange potential (2)

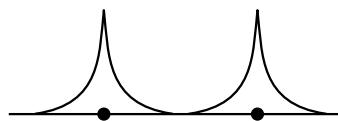


$1/r$ -behavior in  $V_{\text{eff}}$  is missing. (also discuss Helium)

70 / 73

## Lack of dispersion

## LDA and GGA fail to describe London dispersion (1)

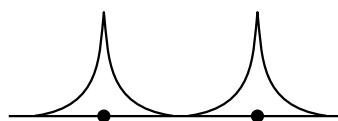


- London dispersion is a long-range  $1/R^6$  attractive interaction between neutral atoms due to correlated motions of electrons in both atoms.
- HF lacks dispersion: no correlation energy.
- DFT is in principle exact but...
- LDA and GGA lack dispersion: mathematical properties of these functionals.

71 / 73

## Lack of dispersion

## LDA and GGA fail to describe London dispersion (2)



**LDA** energy density is a function of  $n$

**GGA** energy density is a function of  $n$ ,  $(\nabla n)^2$  and  $\nabla^2 n$

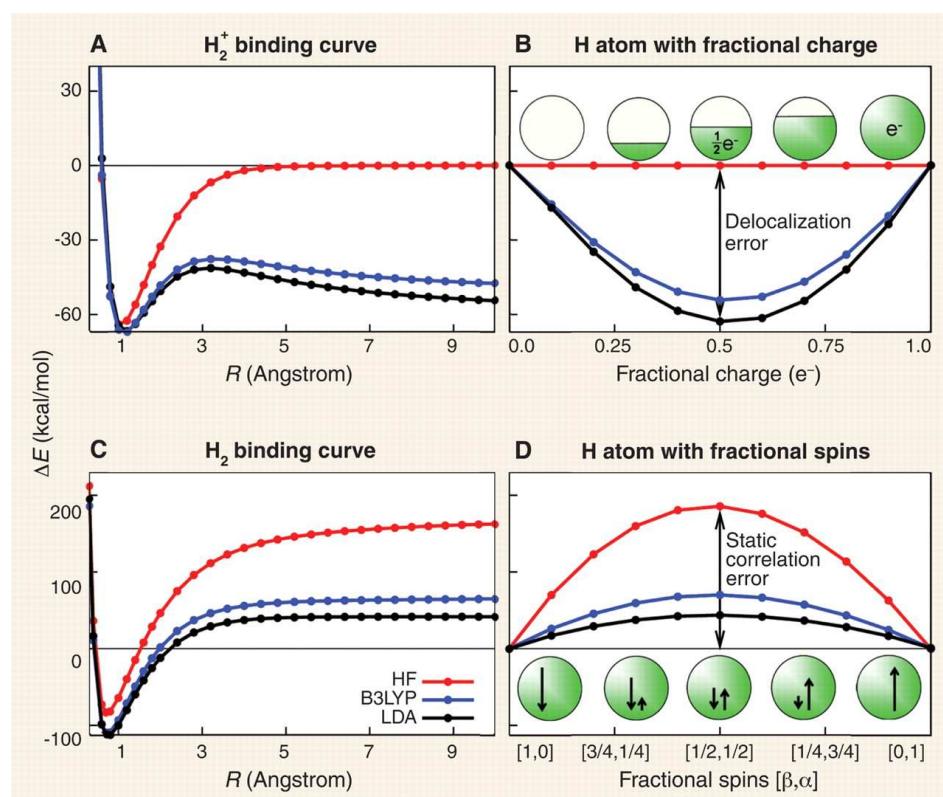
⇒ becomes zero in vacuum

⇒ energy is a single integral of local (LDA) or semilocal (GGA) contributions

⇒ overlap of atomic densities required for interaction energy

72 / 73

# The (de)localization error



Aron J. Cohen, Paula Mori-Sánchez and Weitao Yang, *Science*, 321, 792–794, 2008

73 / 73