Introduction to Octopus:
a real-space (TD)DFT code

Heiko Appel, Martin Lüders, Sebastian Ohlmann, Micael Oliveira,
Nicolas Tancogne-Dejean

Octopus Course 2023, MPSD Hamburg

Installation

- Download the code from:
  `https://octopus-code.org/download/13.0/octopus-13.0.tar.gz`
- Compile:
  - `sudo apt-get install libcgal-dev`
  - `tar xzvf octopus-13.0.tar.gz`
  - `cd octopus-13.0`
  - `mkdir _build`
  - `cd _build`
  - `../configure --prefix=<INSTALLDIR> --enable-openmp`
  - `make -j install`
- The code should now be available as `<INSTALLDIR>/bin/octopus`
- If you want to compile with MPI parallelization, add `--enable-mpi`

Schedule

- Day 1 (11 Sep):
  - Introduction: DFT, TD-DFT and the Octopus code
  - Octopus basics tutorial series
- Day 2 (12 Sep):
  - From the theory to practical numerical implementations (Nicolas Tancogne-Dejean)
  - Optical absorption tutorial series
- Day 3 (13 Sep):
  - Solids tutorial series
  - Model systems tutorial series
- Day 4 (14 Sep):
  - Model systems tutorial series
  - Postopus tutorials
- Day 5 (15 Sep):
  - Introduction to Maxwell systems in Octopus (Heiko Appel)
  - Maxwell tutorials

Theory

A bit of underlying theory: Density Functional Theory

- Hohenberg-Kohn theorem
- Kohn-Sham system
- time-dependent DFT

Density Functional Theory

Hohenberg-Kohn theorem

- One-to-one mapping: $v(\vec{r}) \xleftrightarrow{W} n(\vec{r})$
- All observables are functionals of the density
- Variational principle:

$$\begin{aligned} E_{v_0}[n_0] &= E_0 && \text{for ground state } n_0 \\ E_{v_0}[n] &> E_0 && \text{for} \quad n \neq n_0 \end{aligned}$$

- Universal functional $F[n]$:

$$E_{v_0}[n] = \underbrace{\langle \Psi[n] | \hat{T} + \hat{W} | \Psi[n] \rangle}_{=F[n]} + \int \mathrm{d}^3 r \, n(\vec{r}) \, v_0(\vec{r})$$

- $F[n]$ does not depend on external potential.

Density Functional Theory

Kohn-Sham scheme

- Use one-to-one mapping: define effective non-interacting system

$$v(\vec{r}) \underbrace{\longleftrightarrow}_{W} n(\vec{r}) \underbrace{\longleftrightarrow}_{W\equiv 0} v_{\mathrm{s}}(\vec{r})$$

- Rewrite universal functional:

$$F[n] = T_s[n] + \frac{1}{2}\int \mathrm{d}^3r \int \mathrm{d}^3r'\, n(\vec{r})w(\vec{r},\vec{r}')n(\vec{r}') + E_{xc}[n] \qquad (1)$$

- Apply variational principle (Euler-Lagrange equations): Kohn-Sham equations

Density Functional Theory

Kohn-Sham equations

$$\left[-\frac{1}{2}\nabla^2 + v_{\mathrm{ext}}(\boldsymbol{r}) + v_{\mathrm{H}}[n](\boldsymbol{r}) + v_{\mathrm{xc}}[n](\boldsymbol{r})\right]\psi_i(\boldsymbol{r}) = \epsilon_i\psi_i(\boldsymbol{r})$$

$$n(\boldsymbol{r}) = \sum_{i=1}^{N} |\psi_i(\boldsymbol{r})|^2$$

- A self-consistency scheme is used to treat the non-linearity
- Solve for eigenstates at fixed v_{Hxc}, then update n and v_{Hxc}
- It is a boundary value problem
- Note: Kohn-Sham states are auxiliary quantities. They to not represent excitations of the system!

Time-dependent Density Functional Theory

Runge-Gross theorem

- One-to-one mapping (for given initial state):
  $v(\vec{r},t) + c(t) \overset{W}{\longleftrightarrow} n(\vec{r},t)$
- All observables are functionals of the density and the initial state
- Formulation of a variational principle based on the action is more complicated (causality paradox)

- A time-dependent Kohn-Sham system can be defined in analogy to ground-state DFT

Time-dependent Density Functional Theory

Time-dependent Kohn-Sham equation

$$\mathrm{i}\frac{\partial}{\partial t}\psi_i(\boldsymbol{r},t) = \left(-\frac{1}{2}\nabla^2 + v_{\mathrm{ext}}(\boldsymbol{r},t) + v_{\mathrm{H}}[n](\boldsymbol{r},t) + v_{\mathrm{xc}}[n](\boldsymbol{r},t)\right)\psi_i(\boldsymbol{r})$$

$$n(\boldsymbol{r},t) = \sum_{i=1}^{N} |\psi_i(\boldsymbol{r},t)|^2$$

- It is an initial value problem
- Usually the ground-state is used as initial state
- Various numerical schemes for doing the time-propagation

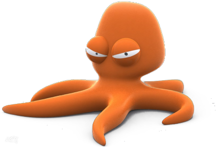
(Time-dependent) Density Functional Theory

Summary

- DFT (Hohenberg-Kohn / Runge-Gross + Kohn-Sham) is in principle exact
- The exact exchange-correlation functionals are unknown
- The xc functionals have to be approximated

The Octopus code

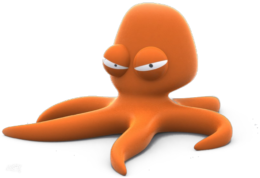
Purpose: simulate the dynamics of electrons and nuclei under the influence of external time-dependent fields in the framework of Time-Dependent Density Functional Theory (TDDFT)



- DFT with many functionals (from `Libxc`), Hartree-Fock, Hartree
- Fortran 2008, C, C++, OpenCL/CUDA and some Python and Perl.
- extensive use of mathematical libraries: `BLAS/LAPACK`, `FFTW`, `GSL`, etc.
- Interfaces to external libraries: `libxc`, `libvdwxc`, `wannier90`, `berkeleygw`, etc.
- Free open-source software (GNU Public License).
- Current version is 13.0.
- Framework to implement and test new ideas

The Octopus code: features and functionalities

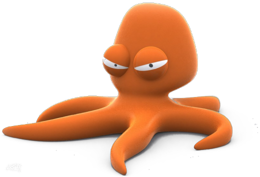
- Ground-state DFT calculations
- Excited states calculations (real-time propagation, linear response)
  - optical (e.g. ARPES)
  - magnetic (magnons)
  - vibrations
- electric and thermal conductivities
- Theory levels include:
  - Kohn-Sham with LDA, GGA
  - generalized Kohn-Sham: MGGA, hybrid functionals
  - Optimized effective potentials (OEP)
  - reduced density matrix functional theory (RDMFT)
  - van der Waals interactions



The Octopus code: new features

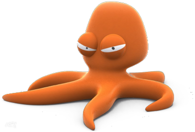
Big changes going on:

- Multi-system mode
  - classical particles (nuclei)
  - electrons
  - Maxwell fields
- Maxwell calculations
  - currently free propagation of $\vec{E}$ and $\vec{B}$ fields, and propagation in linear media.
  - see upcoming tutorial
- Maxwell matter coupling in development
- Move to object oriented design



The Octopus code

- https://octopus-code.org
- https://gitlab.com/octopus-code



M.A.L. Marques, A. Castro, G. F. Bertsch, and A. Rubio, “octopus: a first-principles tool for excited electron-ion dynamics”, *Comput. Phys. Commun.* **151**, 60-78 (2003).

A. Castro, H. Appel, M.J.T. Oliveira, C.A. Rozzi, X. Andrade, F. Lorenzen, M.A.L. Marques, E.K.U. Gross, and A. Rubio, “octopus: a tool for the application of time-dependent density functional theory”, *Phys. Stat. Sol. B* **243**, 2465-2488 (2006).

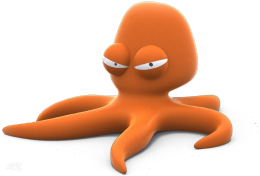
X. Andrade, J. Alberdi-Rodriguez, D.A. Strubbe, M.J.T. Oliveira, F. Nogueira, A. Castro, J. Muguerza, A. Arruabarrena, S.G. Louie, A. Aspuru-Guzik, A. Rubio, and M.A.L. Marques, “Time-dependent density-functional theory in massively parallel computer architectures: the octopus project”, *J. Phys.: Cond. Matt.* **24**, 233202 (2012).

X. Andrade, D.A. Strubbe, U. De Giovannini, A.H. Larsen, M.J.T. Oliveira, J. Alberdi-Rodriguez, A. Varas, I. Theophilou, N. Helbig, M.J. Verstraete, L. Stella, F. Nogueira, A. Aspuru-Guzik, A. Castro, M.A.L. Marques, and A. Rubio, “Real-space grids and the Octopus code as tools for the development of new simulation approaches for electronic systems”, *Phys. Chem. Chem. Phys.* **17**, 31371 (2015).

N. Tancogne-Dejean, M. J. T. Oliveira, X. Andrade, H. Appel, C. H. Borca, G. Le Breton, F. Buchholz, A. Castro, S. Corni, A. A. Correa, U. De Giovannini, A. Delgado, F. G. Eich, J. Flick, G. Gil, A. Gomez, N. Helbig, H. Hübener, R. Jestädt, J. Jornet-Somoza, A. H. Larsen, I. V. Lebedeva, M. Lüders, M. A. L. Marques, S. T. Ohlmann, S. Pipolo, M. Rampp, C. A. Rozzi, D. A. Strubbe, S. A. Sato, C. Schäfer, I. Theophilou, A. Welden, and A. Rubio, “Octopus, a computational framework for exploring light-driven phenomena and quantum dynamics in extended and finite systems”, *The Journal of Chemical Physics* **152**, 124119 (2020)

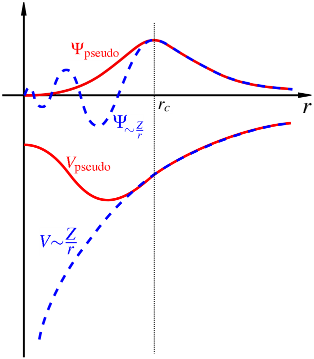
Dissecting the animal

- Real-space grid representation
- Finite differences for the calculation of derivatives
- Pseudopotential approximation
- Finite systems
- Periodic systems
- Arbitrary number of dimensions
- Three methods to obtain excited states properties within TDDFT:
    - Real-time TDDFT
    - Casida Linear response
    - Sternheimer linear response
- Quantum optimal control theory
- Many other features



Pseudo-potentials

- The atomic potential is very strong and "hard" (small spacing or high plane-wave cutoff required).
- Core electrons are strongly bound and do not play a significant role in the chemical binding of atoms.
- Replace the potential and core electrons by a pseudo-potential.



Norm-conserving pseudo-potentials in Kleinman-Bylander form

$$V = V_{\text{loc}} + \sum_{lm} |lm\rangle \left(V_l - V_{\text{loc}}\right) \langle lm|$$

Other potential types

Can be specified in the `Species` block

- all electron potentials (delta, Gaussian or ANC nucleus)
- user defined potentials
- jellium spheres
- and others...

Real-space grid

- We have a partial differential equation with infinite degrees of freedom
- We want a finite number of degrees of freedom
- Discretization: functions are represented by values on a set of points
- Point distribution:
  - Uniformly spaced grid
  - Distance between points is constant: *Spacing*
  - (Non-uniform grids also possible)
- Finite region of the space: *Box*

Real-space grid

- Natural boundary conditions for different problems:
  zero, one, two, or three periodic dimensions for molecules, wires, sheets, and solids
- Can systematically improve discretization quality:
  - Decrease the spacing
  - Increase the box size (in finite directions)
- Orthogonal “basis set”
- Unbiased, independent of atomic positions (no Pulay forces)
- Problems:
  - Breaking of translational invariance: egg-box effect
  - Breaking of rotational invariance
  - (Decreasing spacing helps both)

Discretization of the Hamiltonian

We use finite differences to evaluate the Laplacian (kinetic energy):

- Derivative at a point: weighted sum over neighboring points
- General form for the Laplacian:

$$\nabla^2 f(n_x h, n_y h) = \sum_i^n \sum_j^n \frac{c_{ij}}{h} f(n_x h + ih, n_y h + jh)$$

- The coefficients c_{ij} depend on the mesh and number of points used: *the stencil*
- More points → more precision

The Hamiltonian becomes a finite-size matrix

The eigenvalue problem

- Find the eigenvectors and eigenvalues of a matrix
- Very large matrix with lots of zero components (*Sparse*)
- Use iterative solvers where only the application of the matrix is required (various options available in the code)

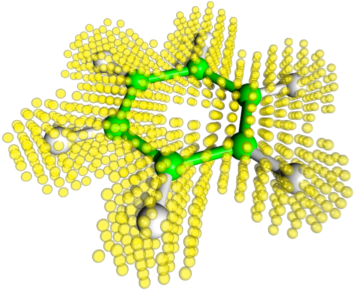
Boundary conditions

- For finite systems, functions go to zero:
  - Force functions to go to zero on the border of the box
  - The box has to be large enough to contain the functions
- Other BCs are possible:
  - periodic
  - zero derivative
  - absorbing
  - etc

Boundary conditions

- Optimize the shape of the box to minimize the number of points needed

- Available box shapes:
  - Sphere
  - Cylinder
  - Parallelepiped
  - Minimum box: union of spheres around each atom
  - Arbitrary (*e.g.* 2D image!)



Benzene molecule in minimal box

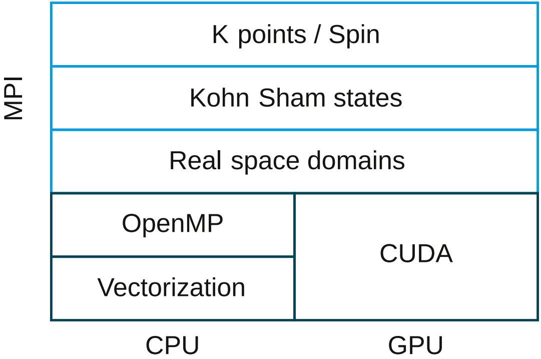
Time propagation

Propagation of the wavefunctions in time:

$$\varphi_i(\boldsymbol{r}, t + \Delta t) = \hat{T}\exp\left\{-\mathrm{i}\int_t^{t+\Delta t} \mathrm{d}t\, \hat{H}\varphi_i(\boldsymbol{r}, t)\right\}$$

- Several numerical methods available for doing the time-propagation
- Exponential must also be calculated numerically
- Many properties can be obtained
- Response to time-dependent fields: lasers
- See e.g. tutorials on optical absorption

Parallelisation strategy



MPI

K points / Spin

Kohn Sham states

Real space domains

OpenMP

Vectorization

CUDA

CPU

GPU

The tutorials

You can find the tutorials under this link:
`https://octopus-code.org/documentation/13/tutorial/`

Have fun!