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

Martin Lüders, Alex Buccheri, Heiko Appel, David Strubbe, Nicolas Tancogne-Dejean

Benasque 2025

# Schedule

- Day 1 (08 Apr):
  - Octopus basics tutorial series
- Day 2 (09 Apr):
  - Optical absorption tutorial series
- Day 3 (10 Apr):
  - Solids tutorial series
- Day 4 (11 Apr):
  - Maxwell tutorials
- Day 5 (13 Apr):
  - BerkeleyGW tutorials

# 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, RDMFT
- 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 16.0.
- Framework to implement and test new ideas

# The Octopus code: features and functionalities

- Ground-state DFT calculations

# The Octopus code: features and functionalities

- Ground-state DFT calculations
- Excited states calculations (real-time propagation, linear response)

# The Octopus code: features and functionalities

- Ground-state DFT calculations
- Excited states calculations (real-time propagation, linear response)
  - optical (e.g. ARPES)

# The Octopus code: features and functionalities

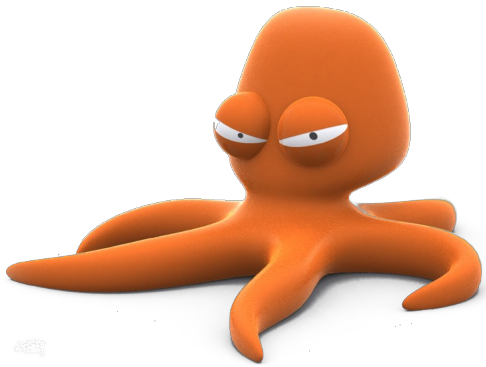
- Ground-state DFT calculations
- Excited states calculations (real-time propagation, linear response)
  - optical (e.g. ARPES)
  - magnetic (magnons)

# 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

# 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



# The Octopus code: features and functionalities

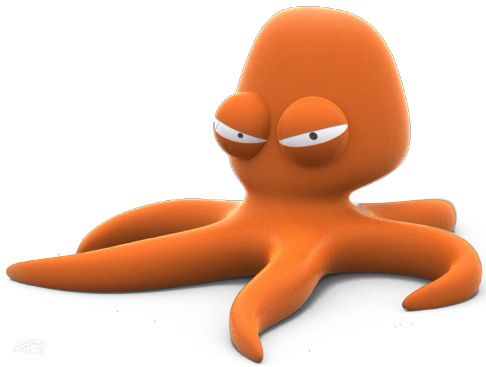
- Ground-state DFT calculations
- Excited states calculations (real-time propagation, linear response)
  - optical (e.g. ARPES)
  - magnetic (magnons)
  - vibrations
- electric and thermal conductivities
- Theory levels include:

# 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

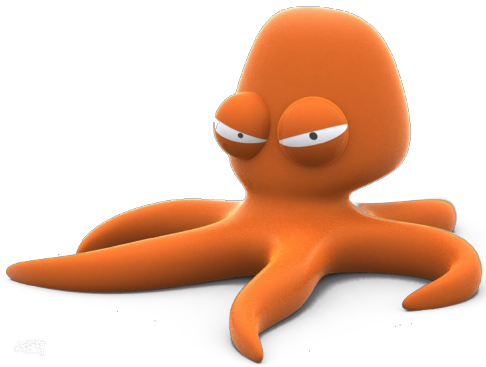
# The Octopus code: features and functionalities

- 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



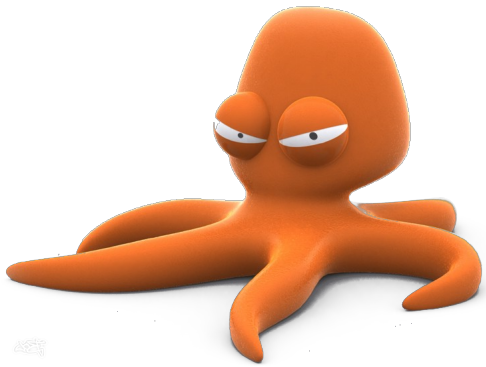
# The Octopus code: features and functionalities

- 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)



# 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)



# 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

# The Octopus code: new features

Big changes going on:

- Multi-system mode
  - classical particles (nuclei)

# The Octopus code: new features

Big changes going on:

- Multi-system mode
  - classical particles (nuclei)
  - electrons

# The Octopus code: new features

Big changes going on:

- Multi-system mode
  - classical particles (nuclei)
  - electrons
  - Maxwell fields

# The Octopus code: new features

Big changes going on:

- Multi-system mode
  - classical particles (nuclei)
  - electrons
  - Maxwell fields
- Maxwell calculations

# The Octopus code: new features

Big changes going on:

- Multi-system mode
  - classical particles (nuclei)
  - electrons
  - Maxwell fields
- Maxwell calculations
  - propagation of $\vec{E}$ and $\vec{B}$ fields, and propagation in linear media.

# The Octopus code: new features

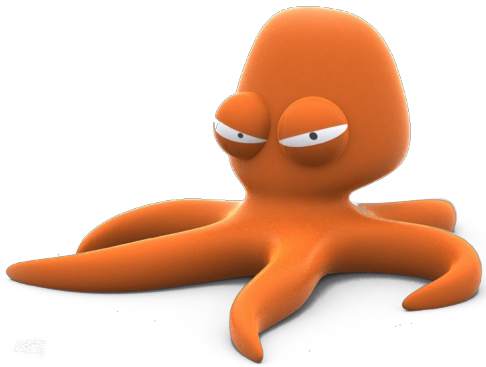
Big changes going on:

- Multi-system mode
  - classical particles (nuclei)
  - electrons
  - Maxwell fields
- Maxwell calculations
  - propagation of $\vec{E}$ and $\vec{B}$ fields, and propagation in linear media.
  - see upcoming tutorial

# The Octopus code: new features

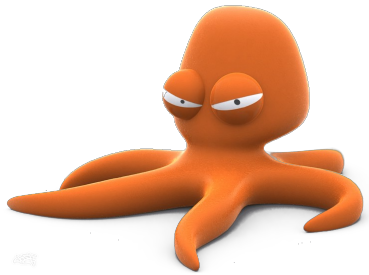
Big changes going on:

- Multi-system mode
  - classical particles (nuclei)
  - electrons
  - Maxwell fields
- Maxwell calculations
  - propagation of $\vec{E}$ and $\vec{B}$ fields, and propagation in linear media.
  - see upcoming tutorial
- 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).

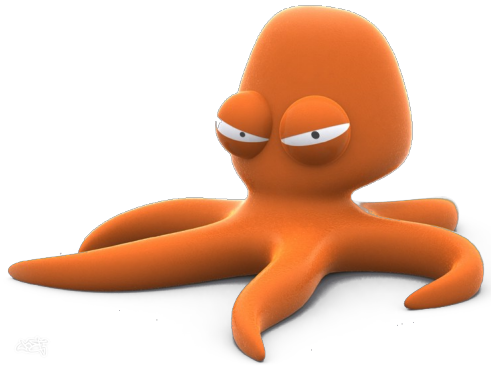
N. Tancogne-Dejean, M. J. T. Oliveira, X. Andrade, H. Appel, C. H. Borca, G. Le Breton, F. Buchholz, A. Castro, S. Corni, A. A. Correa, U. De Giovannini, A. Delgado, F. G. Eich, J. Flick, G. Gil, A. Gomez, N. Helbig, H. Hübener, R. Jestädt, J. Jornet-Somoza, A. H. Larsen, I. V. Lebedeva, M. Lüders, M. 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

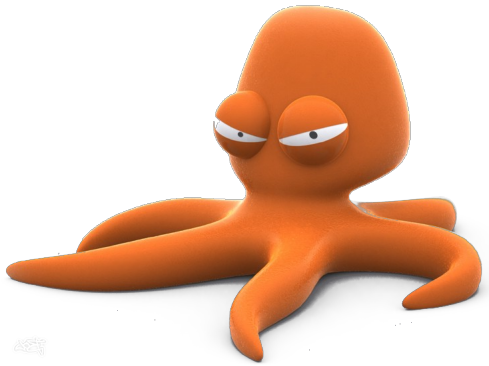
# Dissecting the animal

- Real-space grid representation
- Finite differences for the calculation of derivatives



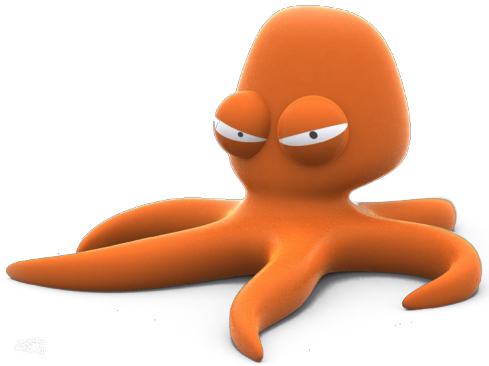
# Dissecting the animal

- Real-space grid representation
- Finite differences for the calculation of derivatives
- Pseudopotential approximation



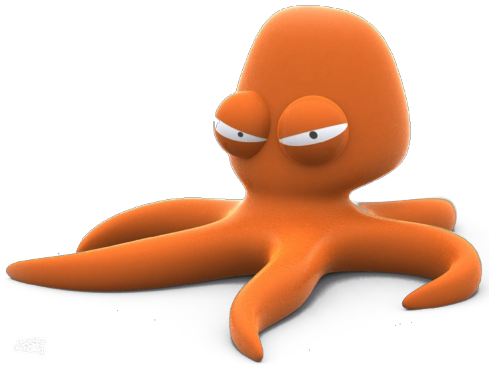
# Dissecting the animal

- Real-space grid representation
- Finite differences for the calculation of derivatives
- Pseudopotential approximation
- Finite systems



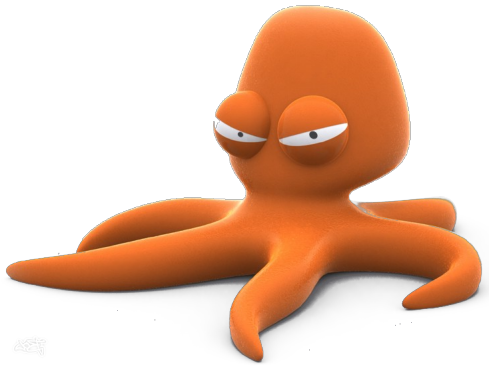
# Dissecting the animal

- Real-space grid representation
- Finite differences for the calculation of derivatives
- Pseudopotential approximation
- Finite systems
- Periodic systems



# Dissecting the animal

- Real-space grid representation
- Finite differences for the calculation of derivatives
- Pseudopotential approximation
- Finite systems
- Periodic systems
- Arbitrary number of dimensions

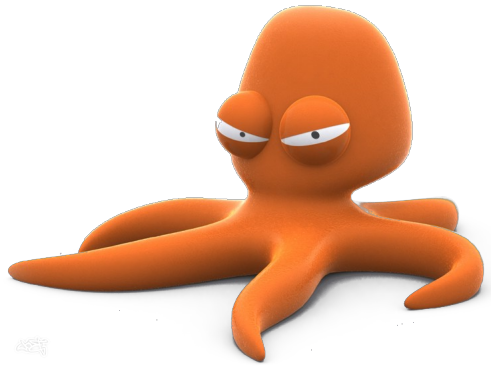


# Dissecting the animal

- 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:

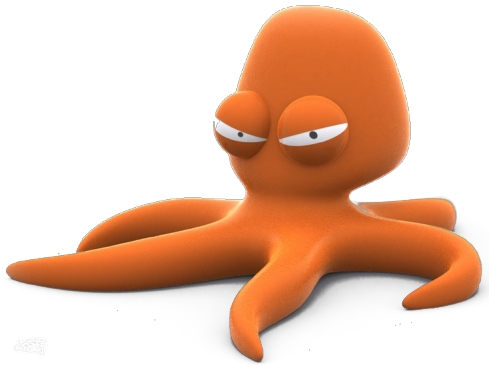
# Dissecting the animal

- 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



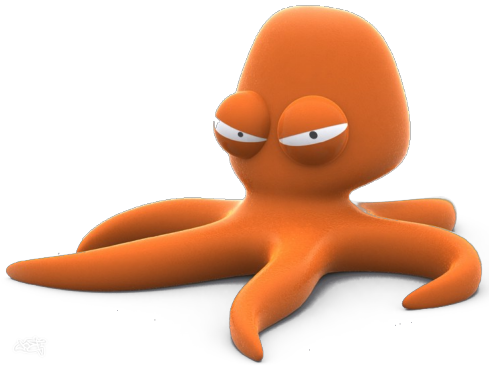
# Dissecting the animal

- 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



# 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



# Dissecting the animal

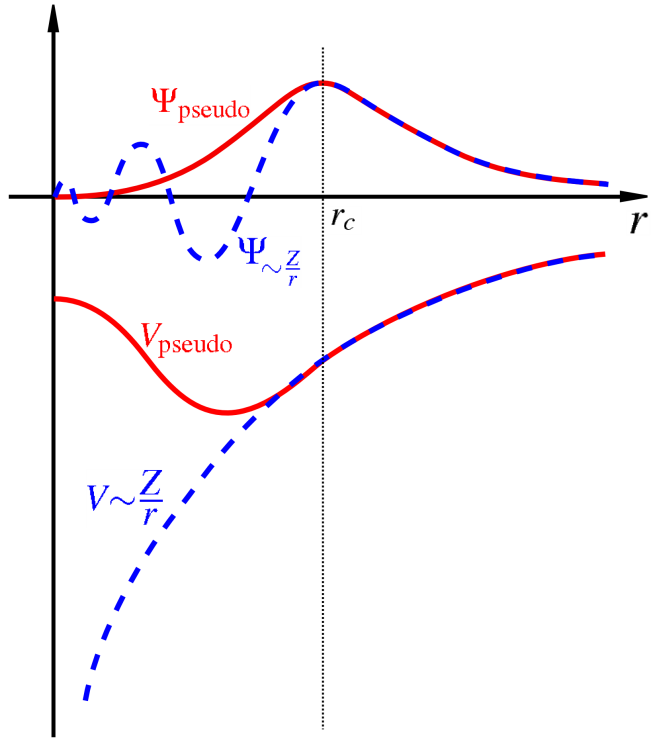
- Real-space grid representation
- Finite differences for the calculation of derivatives
- Pseudopotential approximation
- Finite systems
- Periodic systems
- Arbitrary number of dimensions
- Three methods to obtain excited states properties within TDDFT:
  - Real-time TDDFT
  - Casida Linear response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

# 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

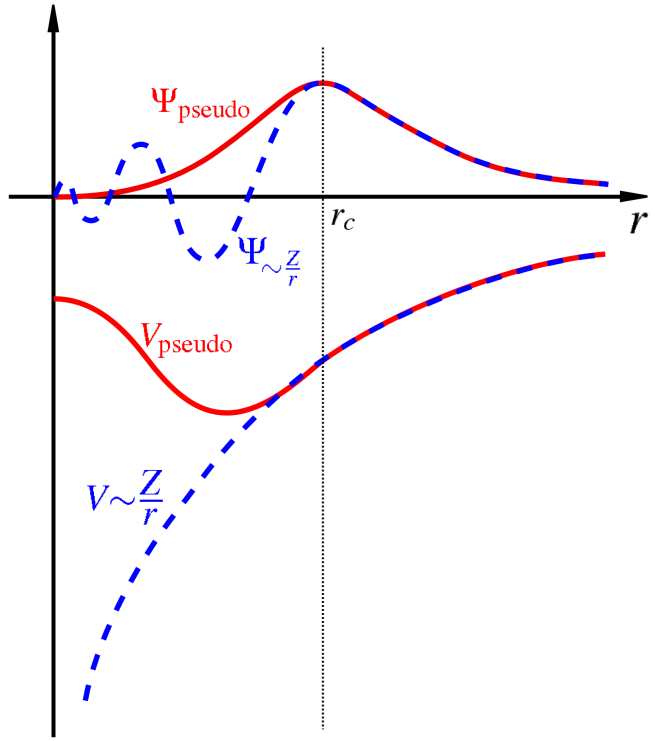
# Pseudo-potentials

- The atomic potential is very strong and "hard" (small spacing or high plane-wave cutoff required).



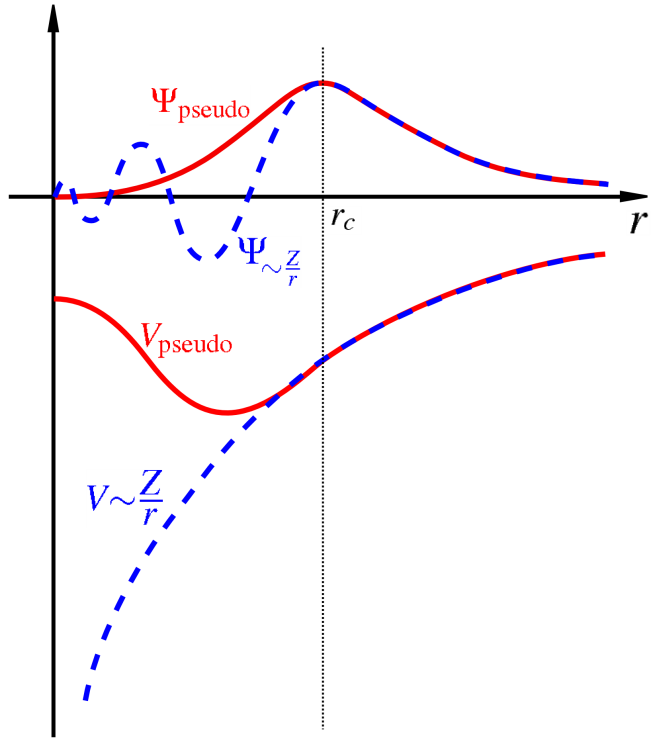
# Pseudo-potentials

- 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.



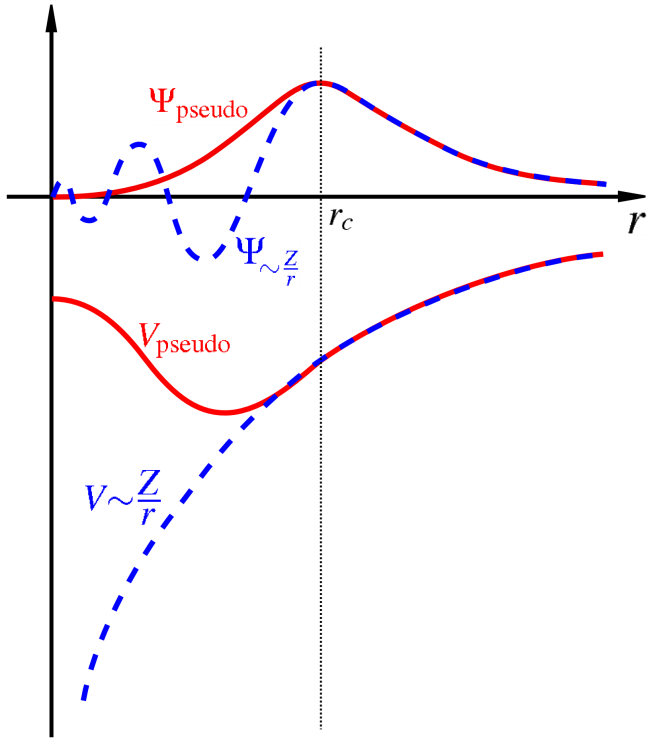
# Pseudo-potentials

- 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.



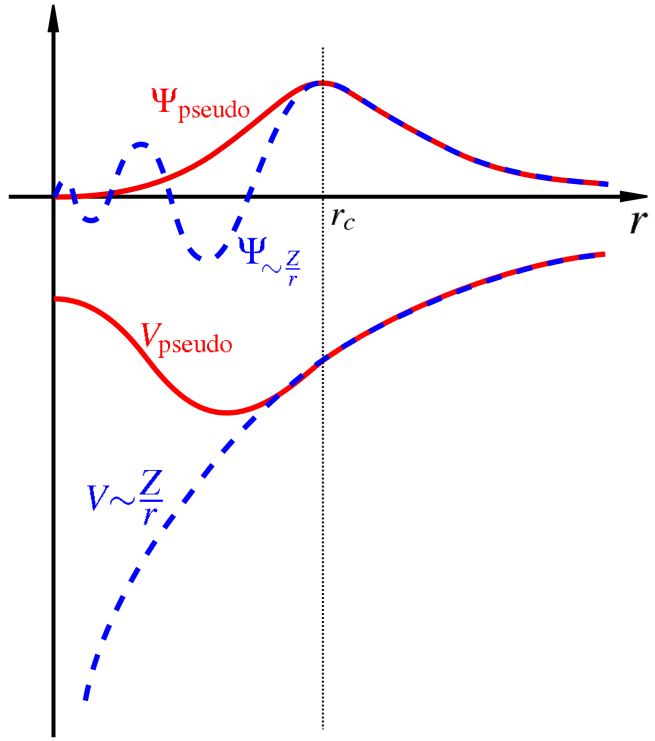
# Pseudo-potentials

- 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.



# 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

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

# Other potential types

Can be specified in the `Species` block

- all electron potentials (delta or Gaussian nucleus)

# Other potential types

Can be specified in the `Species` block

- all electron potentials (delta or Gaussian nucleus)
- user defined potentials

# Other potential types

Can be specified in the `Species` block

- all electron potentials (delta or Gaussian nucleus)
- user defined potentials
- jellium spheres

# Other potential types

Can be specified in the `Species` block

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

# Real-space grid

- We have a partial differential equation with infinite degrees of freedom

# Real-space grid

- We have a partial differential equation with infinite degrees of freedom
- We want a finite number of degrees of freedom

# 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

# 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:

# 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

# 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*

# 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)

# 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

# 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:

# 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

# 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)

# 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"

# 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)

# 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:

# 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

# 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

# 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

# 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:

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

# 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*

# 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 $\rightarrow$ more precision

# 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 $\rightarrow$ more precision

# 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 $\rightarrow$ more precision

The Hamiltonian becomes a finite-size matrix

# The eigenvalue problem

- Find the eigenvectors and eigenvalues of a matrix

# The eigenvalue problem

- Find the eigenvectors and eigenvalues of a matrix
- Very large matrix with lots of zero components (*Sparse*)

# 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:

# Boundary conditions

- For finite systems, functions go to zero:
  - Force functions to go to zero on the border of the box

# 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

# 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:

# 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

# 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

# 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

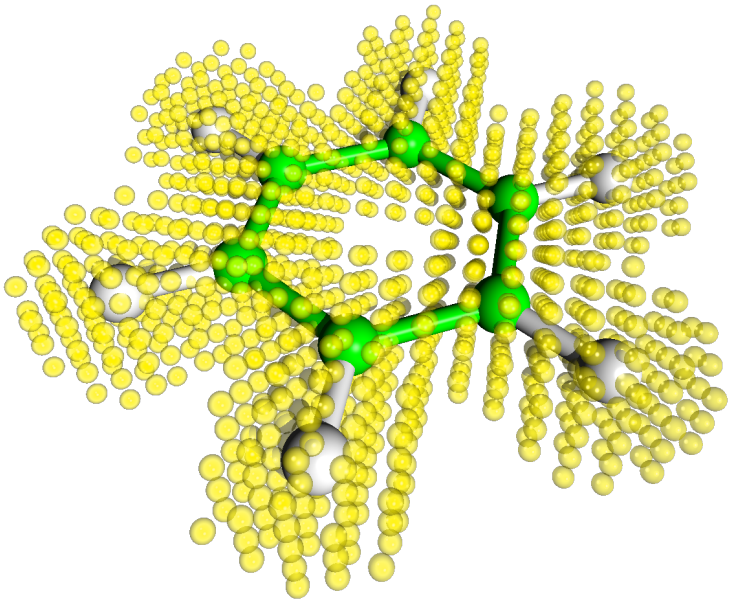
# Boundary conditions

- 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:

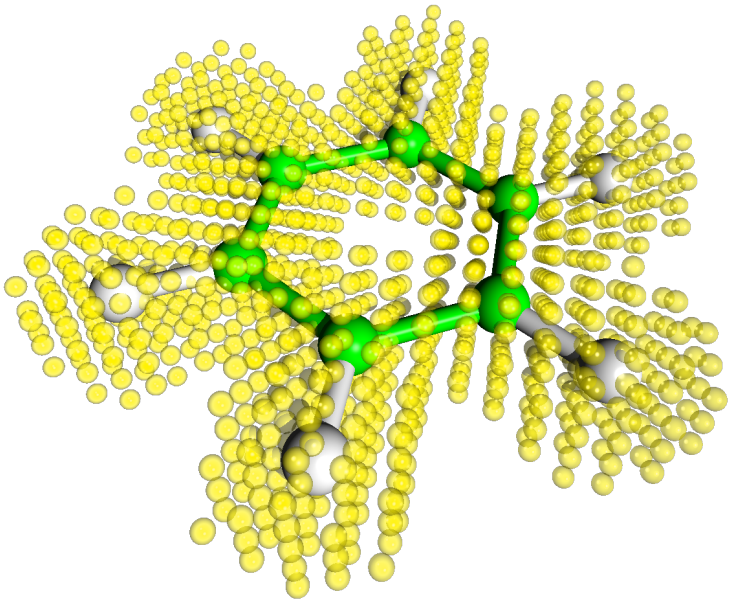


Benzene molecule in minimal box

# Boundary conditions

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

- Available box shapes:
  - Sphere

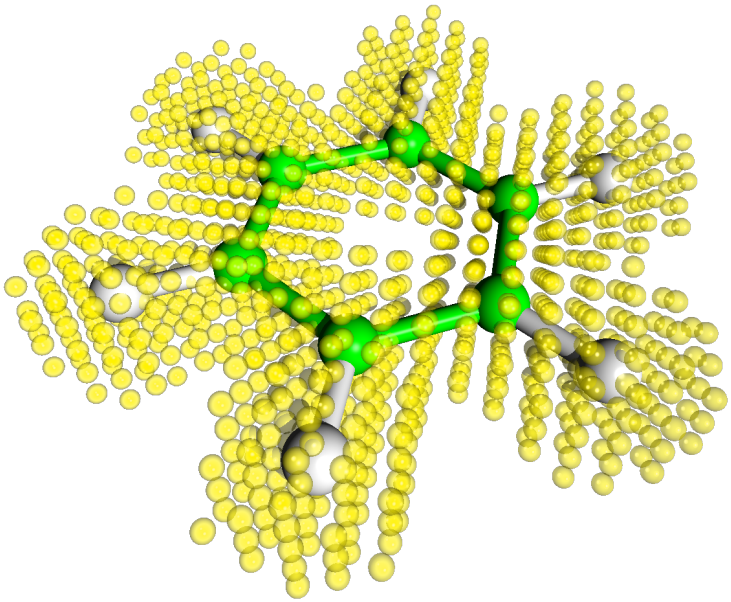


Benzene molecule in minimal box

# Boundary conditions

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

- Available box shapes:
  - Sphere
  - Cylinder

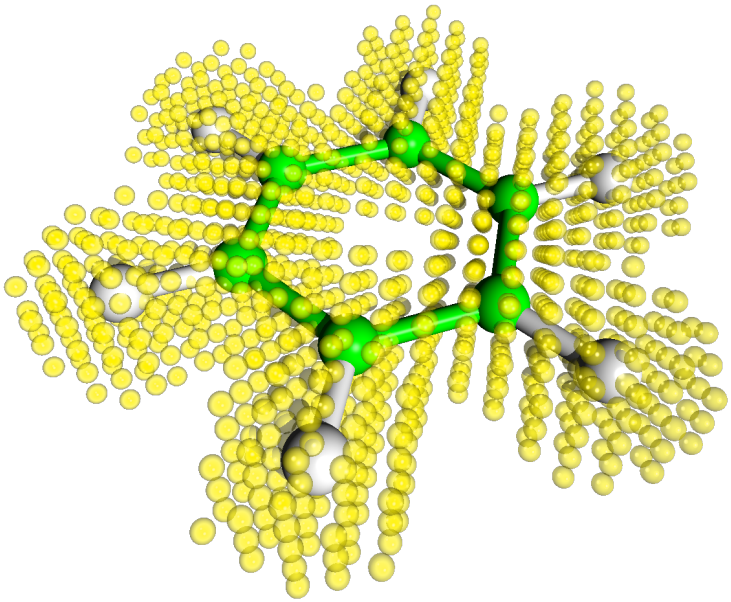


Benzene molecule in minimal box

# Boundary conditions

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

- Available box shapes:
  - Sphere
  - Cylinder
  - Parallelepiped

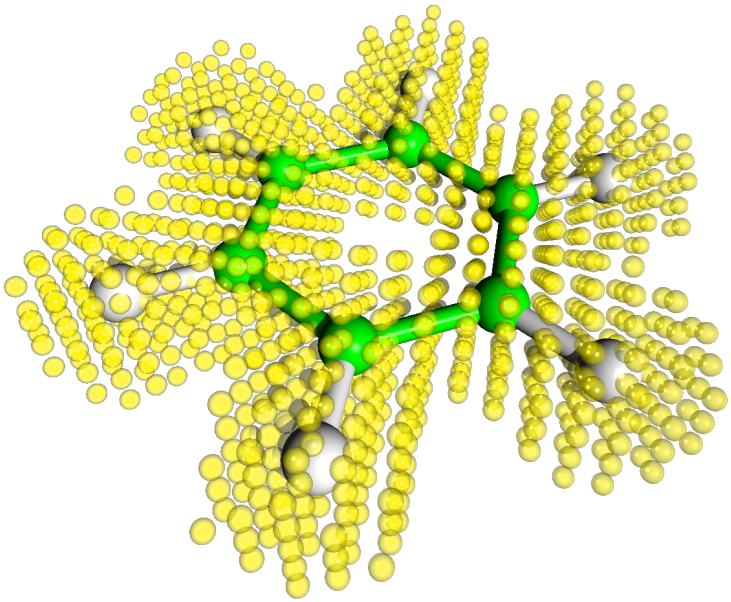


Benzene molecule in minimal box

# 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

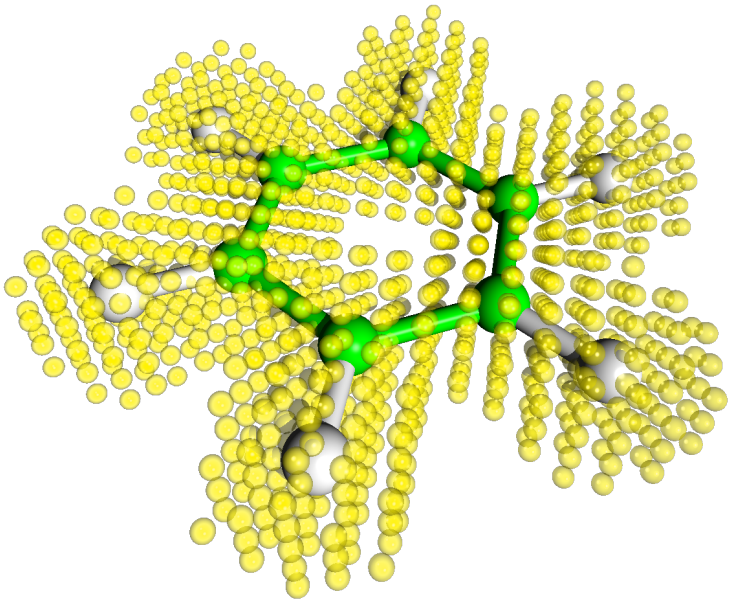


Benzene molecule in minimal box

# 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

# 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

# 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

# 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

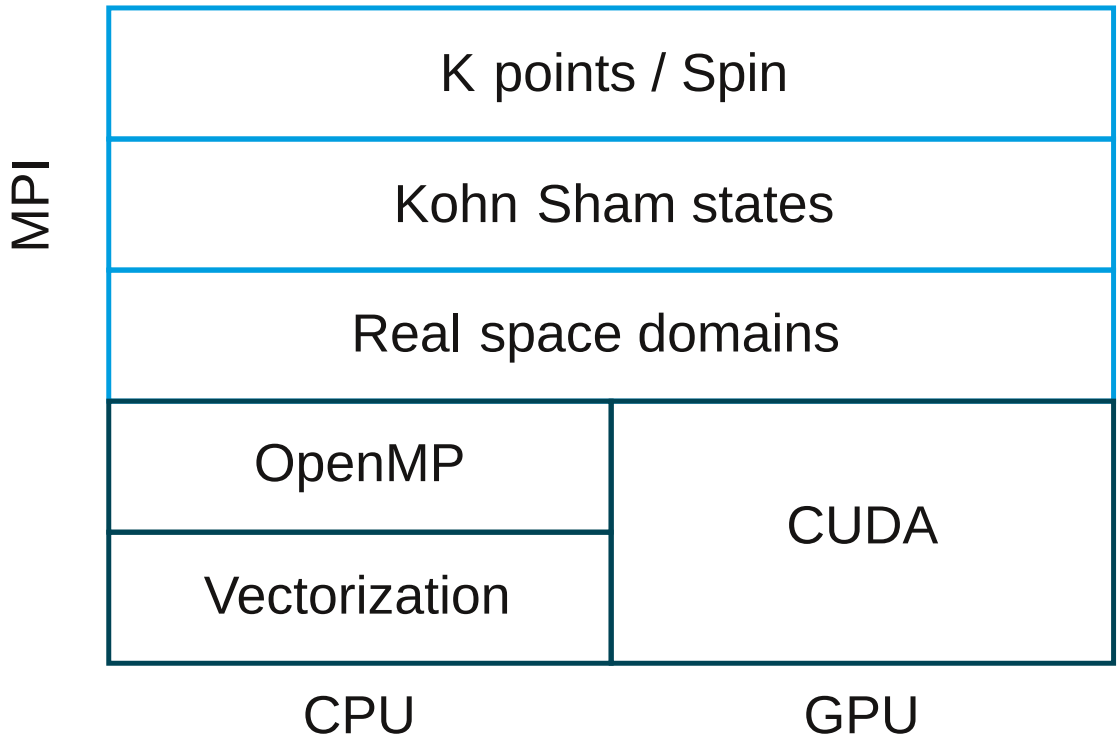
# Time propagation

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



# The tutorials

You can find the tutorials under this link:
`https://www.octopus-code.org/documentation/16/tutorial/`

# The tutorials

You can find the tutorials under this link:
`https://www.octopus-code.org/documentation/16/tutorial/`

Have fun!