Nano-Bio Spectroscopy Group
Universidad del País Vasco

ETSF

Real Space DFT: The gory details

Xavier Andrade

European Theoretical Spectroscopy Facility
and
Departamento de Física de Materiales
Universidad del País Vasco, Spain

Coimbra, April 2008

Objective

- Give an insight of how a real space grid base code works.
- What are the advantages and disadvantages of this method.

Objective

- Give an insight of how a real space grid base code works.
- What are the advantages and disadvantages of this method.

Discretization

- Partial Differential Equation: infinite degrees of freedom.
- Solve it numerically.
- Reduce it to a finite number.
- Retain information.

Discretization

- Partial Differential Equation: infinite degrees of freedom.
- Solve it numerically.
- Reduce it to a finite number.
- Retain information.

Discretization

- Partial Differential Equation: infinite degrees of freedom.
- Solve it numerically.
- Reduce it to a finite number.
- Retain information.

Discretization

- Partial Differential Equation: infinite degrees of freedom.
- Solve it numerically.
- Reduce it to a finite number.
- Retain information.

Real space grid

- Functions are represented by its value over a set of points.
- Point distribution:
  - Uniform space grid.
  - Distance between points is constant: *Spacing*.
  - Non-uniform grids.
- Finite region of the space: *Box*

Real space grid

- Functions are represented by its value over a set of points.
- Point distribution:
  - Uniform space grid.
  - Distance between points is constant: *Spacing*.
  - Non-uniform grids.
- Finite region of the space: *Box*

Real space grid

- Functions are represented by its value over a set of points.
- Point distribution:
  - Uniform space grid.
    - Distance between points is constant: *Spacing*.
    - Non-uniform grids.
- Finite region of the space: *Box*

Real space grid

- Functions are represented by its value over a set of points.
- Point distribution:
  - Uniform space grid.
  - Distance between points is constant: *Spacing*.
  - Non-uniform grids.
- Finite region of the space: *Box*

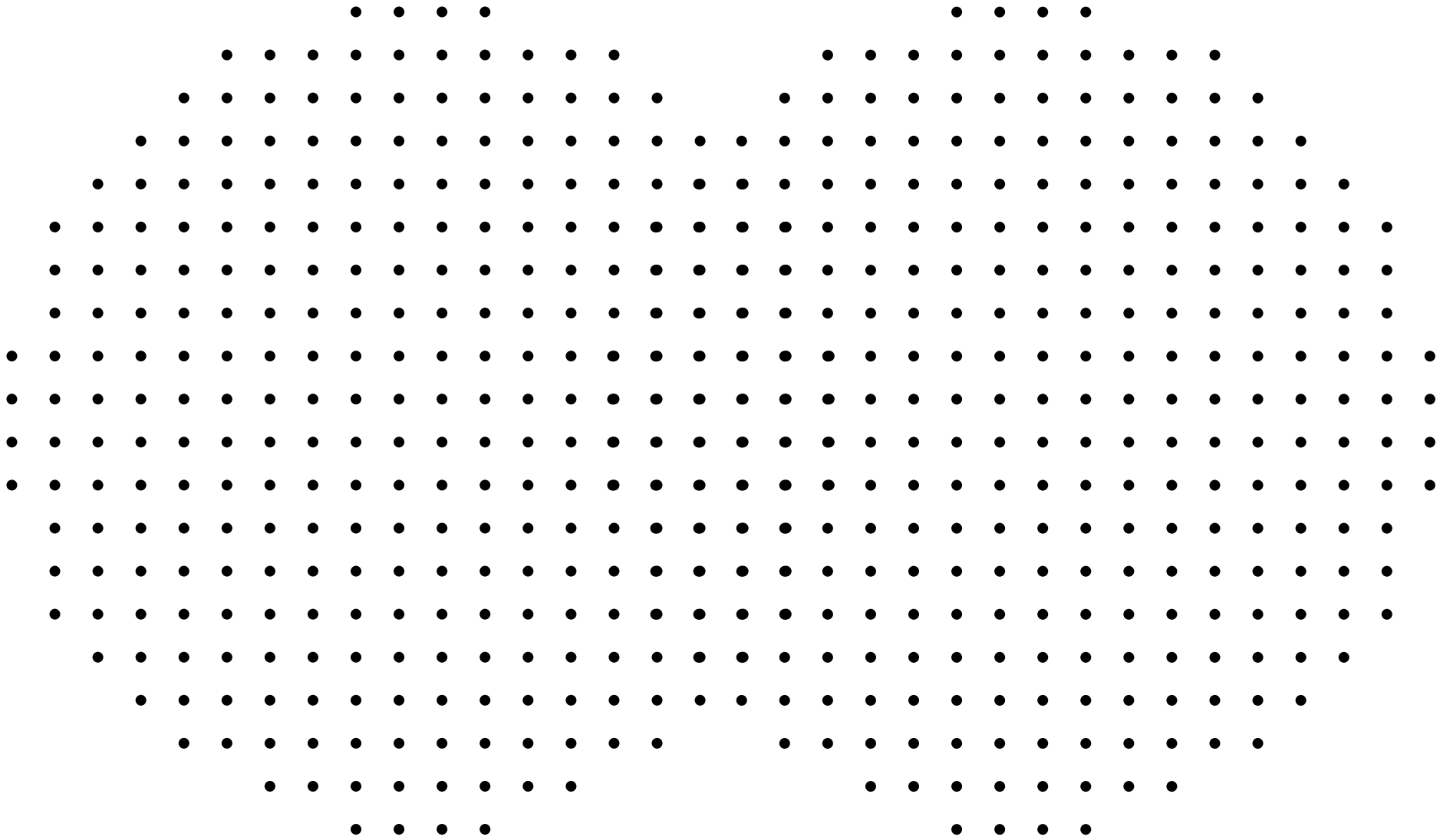
Real space grid

- Functions are represented by its value over a set of points.
- Point distribution:
  - Uniform space grid.
  - Distance between points is constant: *Spacing*.
  - Non-uniform grids.
- Finite region of the space: *Box*

Real space grid

- Functions are represented by its value over a set of points.
- Point distribution:
  - Uniform space grid.
  - Distance between points is constant: *Spacing*.
  - Non-uniform grids.
- Finite region of the space: *Box*

Real space grid: 2D Example



Boundary conditions

- For finite systems functions go to zero.
- Impose functions to be zero over the border of the box.
- The box has to be large enough to contain the functions.
- Optimize the shape of the box to minimize the number of points.
- Other BCs are possible: periodic, zero derivative, open.

Boundary conditions

- For finite systems functions go to zero.
- Impose functions to be zero over the border of the box.
- The box has to be large enough to contain the functions.
- Optimize the shape of the box to minimize the number of points.
- Other BCs are possible: periodic, zero derivative, open.

Boundary conditions

- For finite systems functions go to zero.
- Impose functions to be zero over the border of the box.
- The box has to be large enough to contain the functions.
- Optimize the shape of the box to minimize the number of points.
- Other BCs are possible: periodic, zero derivative, open.

Boundary conditions

- For finite systems functions go to zero.
- Impose functions to be zero over the border of the box.
- The box has to be large enough to contain the functions.
- Optimize the shape of the box to minimize the number of points.
- Other BCs are possible: periodic, zero derivative, open.

Boundary conditions

- For finite systems functions go to zero.
- Impose functions to be zero over the border of the box.
- The box has to be large enough to contain the functions.
- Optimize the shape of the box to minimize the number of points.
- Other BCs are possible: periodic, zero derivative, open.

Real space grid characteristics

- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Real space grid characteristics

- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Real space grid characteristics

- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Real space grid characteristics

- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Real space grid characteristics

- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Real space grid characteristics

- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Real space grid characteristics

- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Real space grid characteristics

- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Real space grid characteristics

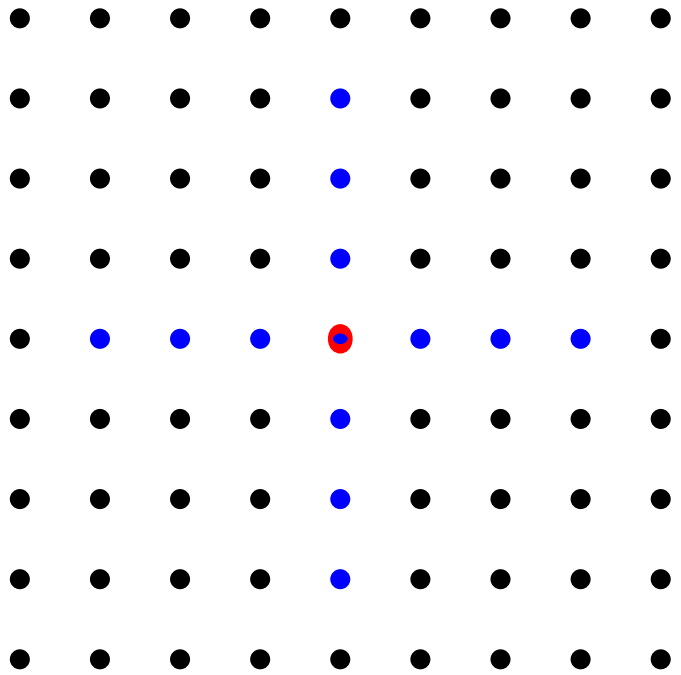
- Natural boundary conditions for different problems.
- Systematically improve discretization quality:
  - Decrease the spacing.
  - Increase the box size.
- Orthogonal “basis set”.
- Independent of atomic positions (no Pulay forces).
- Problems:
  - Break translational invariance: egg-box effect.
  - Break rotational invariance.

Differential operations

Finite difference approach

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

- Derivative in a point: sum over neighbour points.
- c_{ij} depend on the points used: *the stencil*.
- More points $\rightarrow$ more precision.

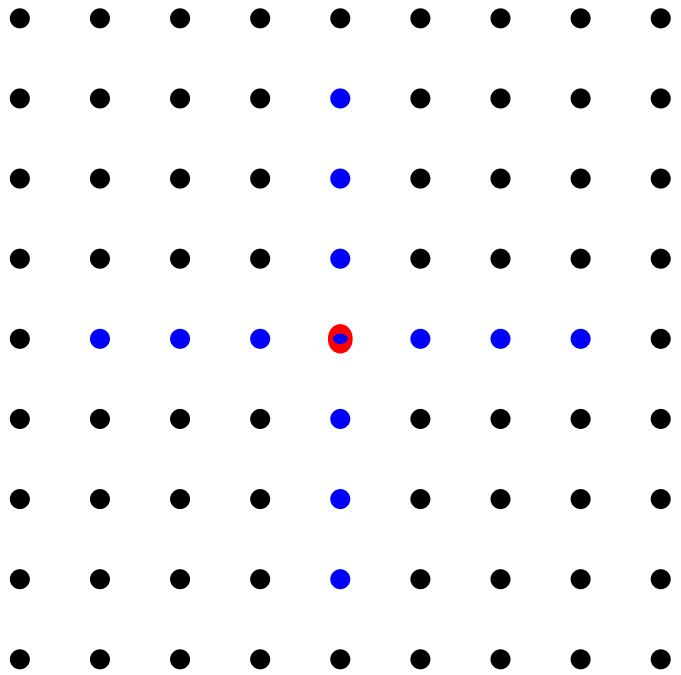


Differential operations

Finite difference approach

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

- Derivative in a point: sum over neighbour points.
- c_{ij} depend on the points used: *the stencil*.
- More points $\rightarrow$ more precision.

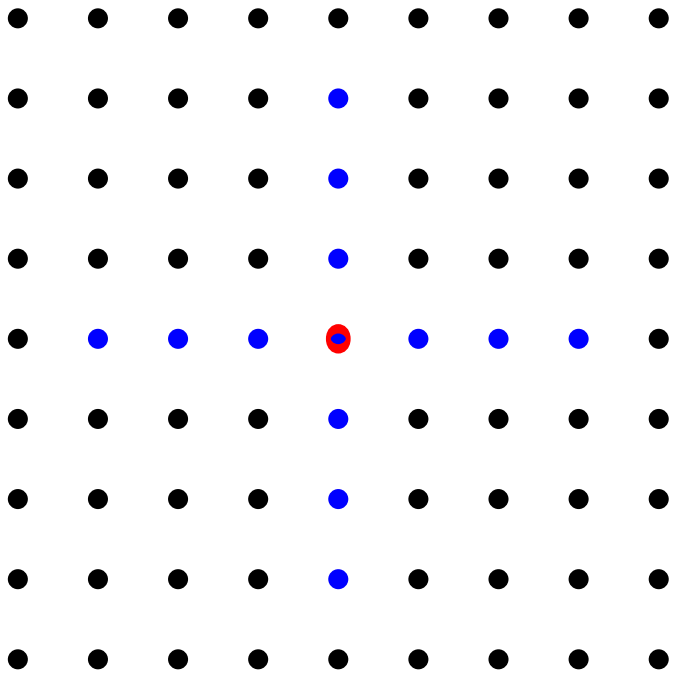


Differential operations

Finite difference approach

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

- Derivative in a point: sum over neighbour points.
- c_{ij} depend on the points used: *the stencil*.
- More points $\rightarrow$ more precision.



Integration

Trapezoidal rule

$$\int f(x,y)\,dx\,dy = h^2 \sum_{ij} f(ih, jh)$$

Ground state calculations

- What we want to solve:

Kohn-Sham equations

$$-\nabla^2\phi_k + V_{eff}\left[\rho\right](\boldsymbol{r})\phi = \epsilon_k\phi_k$$

- We use a self-consistency scheme to treat non-linearity.

Ground state calculations

- What we want to solve:

Kohn-Sham equations

$$-\nabla^2\phi_k + V_{eff}\left[\rho\right](\boldsymbol{r})\phi = \epsilon_k\phi_k$$

- We use a self-consistency scheme to treat non-linearity.

Ground state calculations

- What we want to solve:

Kohn-Sham equations

$$-\nabla^2\phi_k + V_{eff}\left[\rho\right](\boldsymbol{r})\phi = \epsilon_k\phi_k$$

- We use a self-consistency scheme to treat non-linearity.

Discretization of the Hamiltonian

- For the laplacian we use finite differences
  - High order schemes are needed.
- The local part of the potential is direct.
- The non-local potential is applied in small spherical grid around the atoms.
- The Hamiltonian becomes a finite size matrix.

Discretization of the Hamiltonian

- For the laplacian we use finite differences
  - High order schemes are needed.
- The local part of the potential is direct.
- The non-local potential is applied in small spherical grid around the atoms.
- The Hamiltonian becomes a finite size matrix.

Discretization of the Hamiltonian

- For the laplacian we use finite differences
  - High order schemes are needed.
- The local part of the potential is direct.
- The non-local potential is applied in small spherical grid around the atoms.
- The Hamiltonian becomes a finite size matrix.

Discretization of the Hamiltonian

- For the laplacian we use finite differences
  - High order schemes are needed.
- The local part of the potential is direct.
- The non-local potential is applied in small spherical grid around the atoms.
- The Hamiltonian becomes a finite size matrix.

Discretization of the Hamiltonian

- For the laplacian we use finite differences
  - High order schemes are needed.
- The local part of the potential is direct.
- The non-local potential is applied in small spherical grid around the atoms.
- The Hamiltonian becomes a finite size matrix.

The eigenproblem

- We need to find the eigenvectors and eigenvalues of a matrix.
- Very large matrix with lots of zero components (*Sparse*).
- We use Iterative solvers where only the action of the matrix is required.

The eigenproblem

- We need to find the eigenvectors and eigenvalues of a matrix.
- Very large matrix with lots of zero components (*Sparse*).
- We use Iterative solvers where only the action of the matrix is required.

The eigenproblem

- We need to find the eigenvectors and eigenvalues of a matrix.
- Very large matrix with lots of zero components (*Sparse*).
- We use Iterative solvers where only the action of the matrix is required.

The eigensolver

- We minimize (using conjugated gradient or other method):

Rayleigh-Ritz quotient

$$\epsilon(\psi) = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$$

- Works for the first state.
- For higher energy states it is necessary to orthogonalize against the lower ones.

The eigensolver

- We minimize (using conjugated gradient or other method):

Rayleigh-Ritz quotient

$$\epsilon(\psi) = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$$

- Works for the first state.
- For higher energy states it is necessary to orthogonalize against the lower ones.

The eigensolver

- We minimize (using conjugated gradient or other method):

Rayleigh-Ritz quotient

$$\epsilon(\psi) = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$$

- Works for the first state.
- For higher energy states it is necessary to orthogonalize against the lower ones.

The eigensolver

- We minimize (using conjugated gradient or other method):

Rayleigh-Ritz quotient

$$\epsilon(\psi) = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$$

- Works for the first state.
- For higher energy states it is necessary to orthogonalize against the lower ones.

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT
  - TDDFT
    - Time propagation: linear response and strong fields.
    - Casida LR-TDDFT
    - Sternheimer linear and non-linear response
    - Car-Parrinello molecular dynamics
    - Ehrenfest molecular dynamics
    - (adiabatic and non-adiabatic)
    - Optimal control theory
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

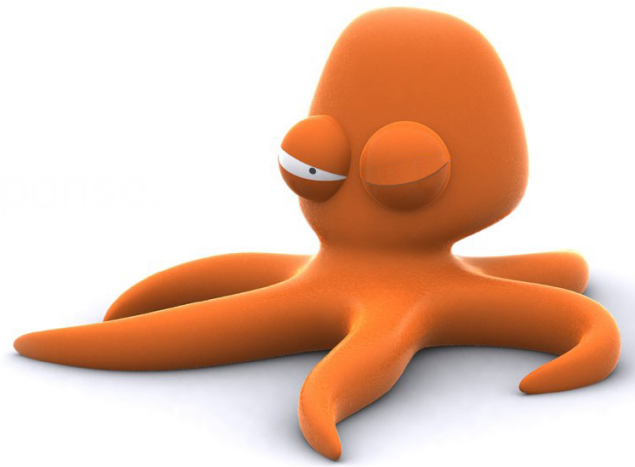
Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
    - Time propagation: linear response and strong fields.
    - Casida LR-TDDFT.
    - Sternheimer linear and non-linear response.
    - Car-Parrinello molecular dynamics.
    - Ehrenfest molecular dynamics (adiabatic and non-adiabatic).
    - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
    - Time propagation, linear response and strong fields.
    - Casida LR-TDDFT.
    - Sternheimer linear and non-linear response.
    - Car-Parrinello molecular dynamics.
    - Ehrenfest molecular dynamics.
    - Adiabatic and non-adiabatic.
    - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.



[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation: linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] `http://www.tddft.org/programs/octopus`

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation: linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus

Octopus[1]

- Open source code.
- Real space grid representation.
- Focused on finite systems (periodic systems not mature yet).
- Features:
  - Ground state DFT.
  - TDDFT
  - Time propagation:
    linear response and strong fields.
  - Casida LR-TDDFT.
  - Sternheimer linear and non-linear response.
  - Car-Parrinello molecular dynamics.
  - Ehrefenst molecular dynamics
    (adiabatic and non-adiabatic).
  - Optimal control theory.
- Norm-conserving pseudopotentials.
- Multilevel parallelization.

[1] http://www.tddft.org/programs/octopus