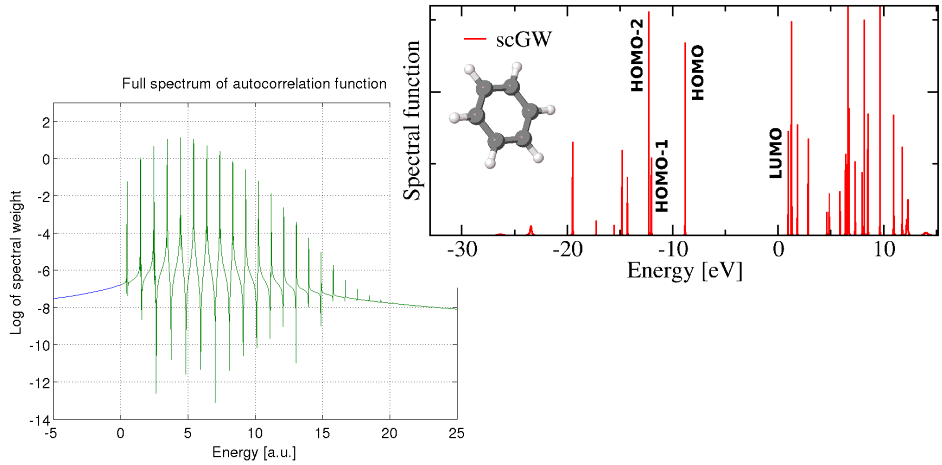
Density functional theory and beyond: Computational material science for real materials

Trieste, August 06 – 15, 2013



Tutorial V: Theoretical Spectroscopy and Electronic Excitations
Manuscript for Exercise Problems

Prepared by Fabio Caruso, Heiko Appel, and Patrick Rinke
Fritz-Haber-Institut der Max-Planck-Gesellschaft, Berlin
August 13, 2013

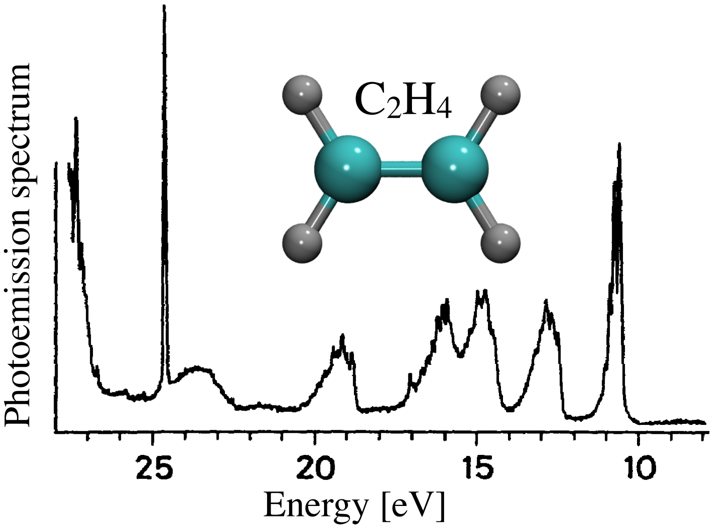
Practical Session V - Theoretical Spectroscopy and Electronic Excitations

This tutorial is divided into two parts. In the first part we will learn about charged electronic excitations. The second part is devoted to the description of neutral electronic excitations.

Part I: Charged electronic excitations

In the first part of this tutorial we will assess the suitability of density-functional theory (DFT), Hartree-Fock (HF) and many-body perturbation theory (MBPT) in the *GW* approach for the calculation of electronic excitations. In this tutorial, most of the calculation will be performed on the ethylene molecule C_2H_4 (and optionally H_2O), with the purpose of comparing the performance of different theoretical approaches with experimental photoemission spectra (illustrated in Fig. 1 for C_2H_4). Hence, we invite you to organise the results of each exercise in a text file, or in the table reported at the end of this document.

Figure 1: Photo-emission spectrum of ethylene adapted from Ref. [9].



A list of experimental ionization energies for C_2H_4 and H_2O can be found for instance in reference [9] and [1], respectively.

Exercise 1: (In)adequacy of DFT eigenvalues for the description of charged electronic excitations

[*Total time for this exercise: 10 minutes. Total CPU time:* < 1 *minute.*]

In this first exercise, you will perform a Kohn-Sham DFT calculation with the PBE exchange-correlation functional and Tier 2 basis set (*tight* settings) for ethylene C_2H_4. The quantities of interest in this case are the Kohn-Sham eigenvalues. You can proceed as follows:

- Create the directory ~`/tutorial5/exercise1`
- Generate the geometry file for ethylene from the experimental data available at `http://cccbdb.nist.gov/` according to the following steps: on the home page of the CCCBDB

database select "Experimental data" from the left column. Follow the link for "Summary of geometry data for one molecule" and provide the molecule name to access the geometry information. Copy the geometry specification to the `geometry.in` file and modify them to comply with the FHI-aims format. As in previous tutorials, the geometry data should be specified in the following format:

```
atom  <X> <Y> <Z> <Element>
atom  <X> <Y> <Z> <Element>
...
```

- Following the template for the `control.in` file available at `$HandsOnDFT/tutorials/tutorial5/reference/PartI/exercise_1_DFT_eigenvalues/C2H4`, run a spin-unpolarized DFT calculation using the PBE exchange correlation functional (flag: `xc pbe`).
- Compare the first four KS eigenvalues – corresponding to the highest occupied molecular orbital (HOMO), HOMO-1, HOMO-2, and HOMO-3 – with the first three experimental ionization energies given in Ref. [2].

Optional: H_2O Repeat the calculation for the water molecule H_2O, using the experimental geometry available at `http://cccbdb.nist.gov/`.

Exercise 2: Electron removal energies from Hartree-Fock.

[*Total time for this exercise: 10 minutes. Total CPU time:* < 1 *minute.*]

Modify the input files of exercise 1 to set up a Hartree-Fock calculation by using the flag

```
xc      hf
```

in the `control.in` file and compare the Hartree-Fock eigenvalues with the experimental ionization energies and with PBE. Note that already for a small molecules such as C_2H_4, the different treatment of exchange and correlation (the latter is absent in HF) may lead to differences in the energy ordering of the orbitals in DFT and HF.

Exercise 3: Electron removal energies from delta-SCF

[*Total time for this exercise: 10 minutes. Total CPU time:* < 1 *minute.*]

In this exercise, the ionization energies (I) of C_2H_4 will be evaluated with the delta-self-consistent-field (Δ-SCF) approach [2]. Following the definition of the ionization potential,

$$I = E_{tot}^{\mathrm{PBE}}(N-1) - E_{tot}^{\mathrm{PBE}}(N) \quad , \tag{1}$$

the total energy difference between the neutral ($E_{tot}^{\mathrm{PBE}}(N)$) and positively ($E_{tot}^{\mathrm{PBE}}(N-1)$) charged species is computed from two separate DFT PBE (or HF) total energy calculations where N is number of electrons of the neutral molecule. Analogously, one can use the Δ-SCF method to evaluate the electron affinity (A) as:

$$A = E_{tot}^{\mathrm{PBE}}(N) - E_{tot}^{\mathrm{PBE}}(N+1) \quad . \tag{2}$$

To evaluate Eq. (1), $E_{tot}^{\mathrm{PBE}}(N)$ can be extracted from the output file of Exercise 1. In addition we need to compute $E_{tot}^{\mathrm{PBE}}(N-1)$, which requires a second DFT calculation. You can proceed as follows:

- Create the directory ∼`/tutorial5/exercise3` and copy the input files from Exercise 1.
- Modify the `control.in` file and set the flag for performing a spin-polarized calculation. Specify the initial spin of the system and the charge of the molecule:

```
xc          pbe
spin         collinear
default_initial_moment  1
charge                 +1
```

Compute the ionization energy and the electron affinity of C_2H_4 using Eq. (1).

- How do these values compare to the bare PBE eigenvalue and to experiment?
- What is the origin of the difference between the Hartree-Fock eigenvalue and the Δ-SCF value for the ionization energy?

Exercise 4: perturbative G_0W_0 and quasi-particle corrections

An improved description of charged electronic excitations is obtained by the perturbative inclusion of many-body effects through the self-energy Σ. In the GW approximation [2] the self-energy is calculated as:

$$\Sigma^{GW}(\mathbf{r},\mathbf{r}',\omega) = \frac{i}{2\pi}\int d\omega' G(\mathbf{r},\mathbf{r}',\omega')W(\mathbf{r},\mathbf{r}',\omega'+\omega) \quad , \tag{3}$$

where $G(\mathbf{r},\mathbf{r}',\omega)$ is the one-particle Green's function and $W(\mathbf{r},\mathbf{r}',\omega)$ is the screened Coulomb interaction (see e.g. Refs. [3,4] for details). The GW self-energy can be used to perturbatively correct the DFT or HF eigenvalues by means of the linearized quasi-particle equation:

$$\epsilon_i^{\mathrm{QP}} = \epsilon_i^{\mathrm{KS}} - \left\langle \psi_i^{\mathrm{KS}} \right| \hat{V}_{\mathrm{xc}}^{\mathrm{KS}} - \hat{\Sigma}_c^{GW}(\epsilon_i^{\mathrm{QP}}) - \hat{\Sigma}_x \left| \psi_i^{\mathrm{KS}} \right\rangle \quad , \tag{4}$$

where Σ_x is the exact-exchange operator, and Σ_c^{GW} is the correlation part of the GW self-energy. $V_{\mathrm{xc}}^{\mathrm{KS}}$ is the exchange-correlation potential of the preliminary calculation, ϵ_i^{KS} and ψ_i^{KS} are the corresponding eigenvalues and eigenvectors, respectively. This approximation is known as G_0W_0 or one-shot GW, because the self-energy is calculated only once, whereas a more rigorous approach would require a fully self-consistent evaluation of Σ. Since the quasi-particle energies in Eq. (4) are evaluated *perturbatively* on top of a preliminary single-particle calculation (generally DFT or Hartree-Fock), the G_0W_0 approach depends on the initial reference calculation. In the following we refer to PBE and Hartree-Fock based G_0W_0 as G_0W_0@PBE and G_0W_0@HF respectively to distinguish between the different starting points.

Exercise 4.1: G_0W_0@PBE quasiparticle energies

[*Total time for this exercise: 10 minutes. Total CPU time:* <1 *minute.*]

The purpose of this exercise is to perform a G_0W_0 calculation for the quasi-particle energies of ethylene.

For this exercise, proceed along the following steps:

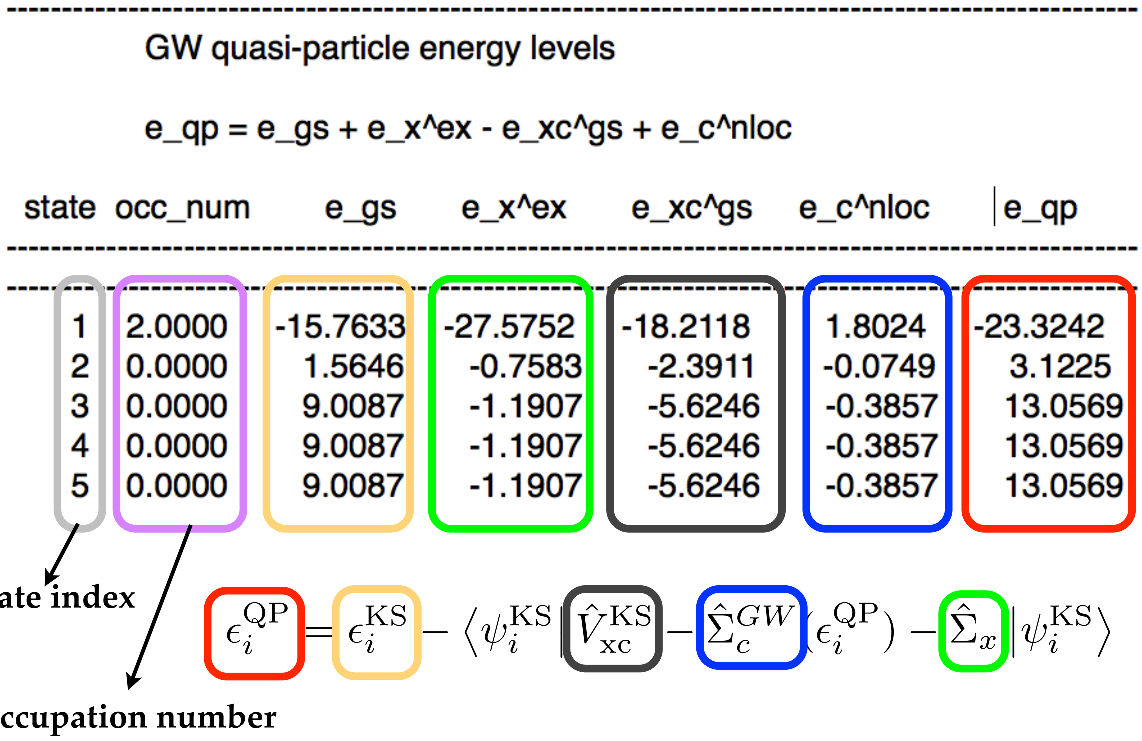
- Create the directory ~`/tutorial5/exercise4` ;
- Copy input files from Exercise 1;
- Modify the `control.in` file including the following flags:

```
xc          pbe
qpe_calc    gw
```

In the species settings at the end of the `control.in` file modify the following flags (for all elements!):

```
cut_pot             6.0   2.0   1.0
basis_dep_cutoff    0
```

Figure 2: Sample output of G_0W_0 quasiparticle calculation in FHI-aims for the hydrogen molecule H_2. The different colors relate terms in the output file to the corresponding quantities in the linearized quasi-particle equation (4).



```
------------------------------------------------------------------------------------------
          GW quasi-particle energy levels

          e_qp = e_gs + e_x^ex - e_xc^gs + e_c^nloc

 state  occ_num       e_gs      e_x^ex     e_xc^gs    e_c^nloc   | e_qp
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
   1    2.0000    -15.7633    -27.5752    -18.2118     1.8024    -23.3242
   2    0.0000      1.5646     -0.7583     -2.3911    -0.0749      3.1225
   3    0.0000      9.0087     -1.1907     -5.6246    -0.3857     13.0569
   4    0.0000      9.0087     -1.1907     -5.6246    -0.3857     13.0569
   5    0.0000      9.0087     -1.1907     -5.6246    -0.3857     13.0569
```

$$\boxed{\epsilon_i^{\mathrm{QP}}} = \boxed{\epsilon_i^{\mathrm{KS}}} - \left\langle \psi_i^{\mathrm{KS}} \middle| \boxed{\hat{V}_{\mathrm{xc}}^{\mathrm{KS}}} - \boxed{\hat{\Sigma}_c^{GW}}(\epsilon_i^{\mathrm{QP}}) - \boxed{\hat{\Sigma}_x} \middle| \psi_i^{\mathrm{KS}} \right\rangle$$

In addition to the output of the DFT PBE calculation, the output file will contain a table – similar to that in Fig.2 – with the quasi-particle corrections to the single-particle eigenvalues. Extract the quasi-particle energies for the HOMO, HOMO-1, HOMO-2, and HOMO-3 levels and compare them with previous results.

Exercise 4.2: G_0W_0 basis set convergence

[*Total time for this exercise: 15 minutes. Total CPU time:* ~ 6 *minutes.*]

Plot the convergence of the first G_0W_0@PBE quasi-particle energy (i.e. the G_0W_0@PBE HOMO level) for ethylene using the Tier 1, Tier 2, and Tier 3 basis sets. Calculations with the Tier 3 basis set will require the following additional settings in the `control.in` file to overcome ill-conditioning of the overlap matrix between basis functions due to the large basis set:

```
basis_threshold 1.e-4
override_illconditioning .true.
```

- How does the convergence of the ionization energy in G_0W_0@PBE compare to that of the PBE eigenvalue?
- What is the origin of the qualitative differences between the convergence behavior in PBE and G_0W_0@PBE?
- Optional: plot the convergence of the first ionization energy of C_2H_4 for `cup_pot`$= 1.0, 2.0, \ldots, 6.0$ with a Tier 2 basis set[1].

Exercise 4.3: G_0W_0@HF and G_0W_0@PBE0: Dependence on the starting point

[*Total time for this exercise: 10 minutes. Total CPU time:* < 1 *minute.*]

The purpose of this exercise is to illustrate the dependence of G_0W_0 calculations on the starting point. Following the steps given in the previous part of this exercise, perform a G_0W_0 calculation using the Hartree-Fock and the PBE0 starting point (i.e. set `xc hf` and `xc pbe0` respectively in the `control.in` file), and compare the HOMO, HOMO-1, HOMO-2, and HOMO-3 quasi-particle energy of the C_2H_4 dimer with G_0W_0@PBE.

(Optional) Exercise 4.4: visualization

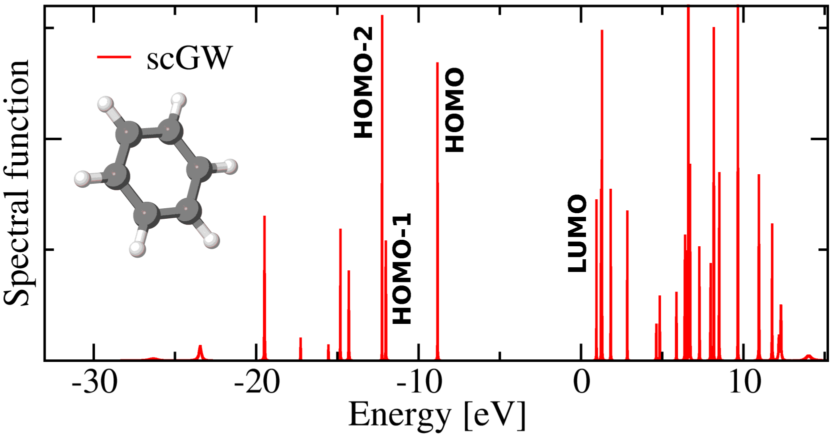
- Extract the G_0W_0 quasi-particle energies from the generated output files and copy them into the files `QPE_HF.dat` and `QPE_PBE.dat` (as a single column of data). The files `QPE_HF.dat` and `QPE_PBE.dat` should contain a single column of numbers. Now run the script `create_spectrum.py` (available in the directory `$HandsOnDFT/tutorials/tutorial5/script/`) in your working directory using the following command:

  ```
  python create_spectrum.py QPE_HF.dat > spectrum_HF.dat
  python create_spectrum.py QPE_PBE.dat > spectrum_PBE.dat
  ```

  The file `spectrum_HF.dat` and `spectrum_PBE.dat` will now contain a spectrum in which the quasi-particle energies were broadened by 0.05 eV to facilitate the comparison with experimental data.
- Visualize (e.g. with `xmgrace`) the two files generated in the previous step of the exercise together with the photo-emission spectroscopy data provided in the file `photo-emission_spectrum_C2H4.dat` located in the directory `$HandsOnDFT/tutorials/tutorial5/reference/PartI/reference_data/`.

[1] Note, that in FHI-aims the default settings for basis sets and integration grids are tuned to optimize the performance of LDA and GGA density functional calculations. Calculations beyond plain DFT, may require the adjustment of such settings.

Figure 3: Example of the spectral function calculated from a self-consistent *GW* Green's function for the Benzene molecule. Since the Green's function has poles at the addition/removal energies of electrons, the position of each peak in the spectral function can be associated with these addition and removal energies.



- How large is the deviation from the experimental HOMO level? How large is the starting point dependence for lower lying quasi-particle energies (e.g. for the HOMO-3 level)?

Exercise 5: Self-consistent *GW*

In this exercise, you will perform a fully self-consistent *GW* calculation. Differently from G_0W_0, the Green's function is calculated by solving the Dyson's equation self-consistently. The Dyson equation relates the input Green's function G_0 to the *GW* Green's function G via the self-energy Σ

$$G(1,2) = G_0(1,2) + \int d34 G_0(1,3)[v_{\rm H}(3)\delta(3,4) + \Sigma(3,4)]G(4,2) \quad , \tag{5}$$

or in inverted form

$$G^{-1}(1,2) = G_0^{-1}(1,2) - v_{\rm H}(1)\delta(1,2) - \Sigma(1,2) \quad , \tag{6}$$

where we used the shortened notation $1 \equiv \{\mathbf{r}_1, t_1, \sigma_1\}$ – see e.g. ref.[4] for an introduction. $v_{\rm H}$ is the Hartree potential. We refer to [7] for details of the sc*GW* implementation in FHI-aims.

Exercise 5.1: Spectral function from the self-consistent Green's function

[*Total time for this exercise: 15 minutes. Total CPU time:* ~ 3 *minutes.*]

To perform a self-consistent *GW* calculation for C_2H_4, create a new directory and copy the input files from Exercise 1. Modify the first part of the *control.in* file:

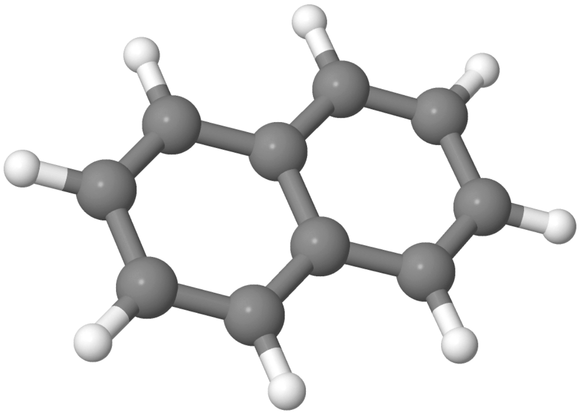
```
xc                pbe
sc_self_energy    scgw
spin              none
```

and **choose Tier 1** settings at the bottom of the `control.in` file.

After running FHI-aims, the file `spectrum_sc.dat` will be created [2] in addition to the usual output files.

[2] Note, that the choice of a spin polarized calculation will produce a spin-resolved spectral function –identical for closed shell systems, such as C_2H_4 and H_2O – named `spectrum_sc_up.dat` and `spectrum_sc_do.dat` for each component of the spin moment.

Figure 4: The Naphthalene molecule.



The file `spectrum_sc.dat` contains the spectral function calculated from the self-consistent GW Green's function

$$A(\omega) = -\frac{1}{\pi} \int \lim_{\mathbf{r}' \to \mathbf{r}} ImG(\mathbf{r}, \mathbf{r}', \omega) d\mathbf{r} \tag{7}$$

where G has been determined self-consistently from Eq. (5). Figure 3 reports an example of a self-consistent GW spectral function for benzene. You can visualize the spectral function using `xmgrace`, `gnuplot` or any other available plotting tool[3]. The first three ionization energies of C_2H_4 must be extracted directly from the spectral function (like in experiment) by reading of the peak positions.

Optional

- Visualization: Using the data collected in the previous exercises for the three highest occupied states (HOMO, HOMO-1, HOMO-2, and HOMO-3) of C_2H_4, visualize in a plot the deviation from the experimental ionization energy for DFT, delta-SCF, Hartree-Fock, G_0W_0@PBE and self-consistent GW.
- Independence on the starting point at self-consistency: in a different folder, perform a second self-consistent GW calculation for C_2H_4 choosing Hartree-Fock as starting point of the calculation and compare the spectral functions obtained in the previous exercise, in which a PBE starting point was used.

Bonus exercise 1: *GW* and the self-interaction error

(Semi-)local functionals, such as LDA or PBE, suffer from the self-interaction error – the incomplete removal of the interaction of an electron with itself introduced in the Hartree term. The self-interaction error is particularly large for localized states, but plays a minor role for delocalized states. Molecules that have both localized and delocalized states that are close in energy (as for instance aromatic molecules), are particularly problematic for LDA and PBE. In such systems, the self-interaction error affects the localized and delocalized states differently, potentially leading to a wrong energetic ordering of the single particle orbitals. In this exercise we illustrate how G_0W_0 establishes the correct energetic ordering by means of a proper treatment of exact exchange in the GW self-energy in Eq. (3).

Perform a G_0W_0 calculation based on an LDA calculation for the naphthalene molecule ($C_{10}H_8$) using the `geometry.in` and control.in available in the directory `$HandsOnDFT/tutorials/tutorial5/reference/PartI/bonus_exercise_Naphthalene/`.

[3] The gnuplot script `$HandsOnDFT/tutorials/tutorial5/reference/PartI/exercise_5_scGW/C2H4/plot.plt` (run the command `gnuplot plot.plt` after copying the script in your working directory) automatically generates a plot of the spectral function.

- Compare the energetic ordering of DFT orbitals and the G_0W_0 quasi-particle energies of naphthalene.
- Plot the orbitals 27 and 28 using the following settings in the control.in file:

  ```
  output cube eigenstate  27
  output cube eigenstate  28
  ```

  then run the script `$HandsOnDFT/tutorials/tutorial5/reference/PartI/bonus_exercise_Naphthalene/plot-eigenstate.sh` in your working directory to produce `png` images of the Kohn-Sham eigenstates.
- How many orbitals are energetically swapped in G_0W_0@LDA compared to LDA?
- Which orbital is more localized: number 27 or number 28?
- Is the different localization of orbitals 27 and 28 consistent with the removal of the self-interaction error and the new energetic ordering?

Bonus exercise 2: *GW* total energy from the Galitskii-Migdal formula

As shown in exercise 5, the single-particle Green's function allows one to determine the energy of single-particle excitations. However, the Green's function may also yield information about the ground state of a system, as it allows to derive, for instance, the total energy and the electron density. To illustrate this aspect, the purpose of this exercise is to calculate the potential energy curve of the hydrogen molecule H_2 using the Galitskii-Migdal formula in the *GW* approximation. The Galitskii-Migdal total energy is an explicit functional of the single-particle Green's function that can be expressed as:

$$E_{\mathrm{GM}} = -i \int \frac{d\omega}{2\pi} Tr\left\{[\omega + h_0]\, G(\omega)\right\} + E_{\mathrm{ion}}, \tag{8}$$

To evaluate the potential energy curve of H_2 you may proceed according to the following steps:

1. Copy the `control.in` from the directory `$HandsOnDFT/tutorials/tutorial5/reference/PartI/bonus_exercise_GW_total_energy/GM` in your working directory. Run self-consistent *GW* calculations for several values of the bond length of H_2 (set `xc hf` in the `control.in`). A suggested set of bond lengths is

   $$d = \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.3, 1.6, 1.9, 2.5, 3.0\}. \tag{9}$$

   We suggest to generate a script for running the calculations. This is a template (available in the directory `$HandsOnDFT/tutorials/tutorial5/reference/PartI/bonus_exercise_GW_total_energy/GM`) for a bash script that could be of aid for speeding up the calculations:

   ```
   #!/bin/bash
   for dist in <set_of_distances> ; do
     echo Running distance  $dist
     echo atom 0.00 0.00 0.00  H  > geometry.in
     echo atom 0.00 0.00 $dist H >> geometry.in
     mpirun -np 4 aims.x \
         > H2_dist$dist.out
   done
   ```

2. Visualize the self-consistent *GW* and the G_0W_0 total energy (evaluated using the Galitskii-Migdal formula) as function of the bond length. The script `$HandsOnDFT/tutorials/tutorial5/reference/PartI/bonus_exercise_GW_total_energy/GM/read_etot.sh` can be used to extract the total energy from the output file of a sc-*GW* calculation.
3. Compare the results with the exact data obtained by full configuration-interaction (full-CI) calculations. The full-CI curve is available in the file

`$HandsOnDFT/tutorials/tutorial5/reference/PartI/reference_data/H2_full-CI.ascii`.

Part II - Neutral electronic excitations

In the previous exercises we have seen how the Kohn-Sham eigenvalues can be used as starting point for the calculation of charged electronic excitations, i.e. excitations where electrons are removed from the system or added to the system. In this part of the tutorial we focus on *neutral electronic excitations* which arise when the system is perturbed by e.g. external electromagnetic fields. As the name of the excitations already indicates, we refer to *neutral electronic excitations* when the particle number of the considered system remains constant during the excitation. This includes processes as e.g. exciton formation in solar cells, absorption of photons in photosynthesis and the vision process in the eye.

The tutorial consists of two parts: The first part gives a brief introduction to the computation of neutral excitation energies in the TDDFT/Casida approach. We investigate vertical Franck-Condon transitions in the C_2H_4 molecule at the equilibrium geometry and compute excited Born-Oppenheimer surfaces along the twisting angle of the molecule. The second part introduces several real-time evolution algorithms which allow to propagate electronic or nuclear wave packets in real-time. The purpose of the second part is to familiarize you with wavepacket propagations in real-time and to compare the stability and efficiency of different propagation algorithms. To that end we employ a few `Matlab`/Octave[4] scripts which easily allow to investigate the structure of the propagation algorithms. Once we are familiar with real-time propagations we employ the ground and excited state potential energy surfaces of the first part to study the non-adiabatic nuclear wave packet evolution on the lowest two coupled Born-Oppenheimer surfaces of C_2H_4.

Neutral excitations in TDDFT - Casida equation

For this tutorial we consider the Casida equation [9] which allows to compute neutral excitation energies as eigenvalues of an algebraic eigenvalue problem

$$\hat{W} F_q = \Omega_q^2 F_q. \tag{10}$$

Here, Ω_q denote the excitation energies of the system, the eigenvectors F_q contain information about the oscillator strength of the considered optical transition and the matrix W is given in terms of Kohn-Sham eigenvalue differences and matrix elements of Kohn-Sham orbitals

$$W_{qq'} = \delta_{qq'}(\epsilon_a - \epsilon_i)^2 + 2\sqrt{(\epsilon_a - \epsilon_i)}\, K_{qq'}(\Omega_q)\sqrt{(\epsilon_{a'} - \epsilon_{i'})}, \qquad q = (a, i). \tag{11}$$

The composite indices q, q' denote pairs $(a, i), (b, j)$ of occupied i, j and virtual a, b orbitals and the matrix $K_{qq'}$ contains matrix elements of the Hartree-exchange correlation kernel $f_{\text{HXC}}(\mathbf{r}, \mathbf{r}', \omega)$ of TDDFT

$$K_{ai,bj}(\omega) = \int\int \psi_a(\mathbf{r})\psi_i^*(\mathbf{r}) f_{\text{HXC}}(\mathbf{r}, \mathbf{r}', \omega)\psi_b(\mathbf{r}')\psi_j^*(\mathbf{r}') d^3r\, d^3r'. \tag{12}$$

In the present tutorial we use the adiabatic LDA kernel $f_{\text{HXC}}^{\text{ALDA}}$, which takes the form

$$f_{\text{XC}}^{\text{ALDA}}[n_0](\mathbf{r}, t, \mathbf{r}', t') = \delta(t - t')\delta(\mathbf{r} - \mathbf{r}')\frac{d^2}{d\rho^2}\left(\rho e_{\text{xc}}^{\text{hom}}(\rho)\right)|_{\rho=n_0(\mathbf{r})}. \tag{13}$$

Here, $e_{\text{xc}}^{\text{hom}}(\rho)$ denotes the exchange-correlation energy density of the uniform electron gas.

[4] Matlab is a powerful commerical package tailored for numerical computations. The package uses a own scripting language which has similarities to Fortran and C. GNU Octave is a open source implementation that can be used to execute Matlab codes

Exercise 6: Vertical Frank-Condon transitions

[*Total time for this exercise: 10 minutes. Total CPU time:* < 1 *minute.*]

In this exercise, we consider vertical Frank-Condon excitations for the ethylene molecule C_2H_4 at the equilibrium geometry. You can take the following steps

- Create a directory ∼`/tutorial5/exercise6`
- Copy your `geometry.in` and `control.in` file from `$HandsOnDFT/tutorials/tutorial5/reference/PartI/exerci`
- In order to prepare a Casida calculation with FHI-aims, you have to add the following lines to the `control.in` file

  ```
  neutral_excitation tddft
  tddft_kernel libxc
  tddft_x XC_LDA_X
  tddft_c XC_LDA_C_PW
  excited_states 20
  excited_mode singlet
  empty_states          10000
  ```

  The first keyword `neutral_excitation tddft` activates the TDDFT/Casida module in FHI-aims. With the next three keywords `tddft_kernel, tddft_x, tddft_c` you select the adiabatic LDA kernel from the libxc library, cf. Ref. [10]. With the keyword `excited_states` you can specify how many excited states FHI-aims should include in the output and with `excited_mode singlet` you can select only singlet excitations. Finally, the keyword `empty_states` specifies how many unoccpuied states FHI-aims should use in order to build the Casida matrix. To include all states this variable is typically set to a very large number.
- Once you prepared the input run FHI-aims

In the output of FHI-aims you can see the lowest neutral singlet excitation energies of the molecule

```
| Writing Singlet excitation energies (TDDFT)
|    1. Singlet:      6.9424 eV     [  0.2551 Ha]  -  Oscillator Strength:       0.0991
|          Transition Moments    X:   -0.3855   Y:    0.0000   Z:    0.0000
|
|    2. Singlet:      7.4981 eV     [  0.2756 Ha]  -  Oscillator Strength:       0.0000
|          Transition Moments    X:    0.0000   Y:    0.0000   Z:    0.0000
|
|    3. Singlet:      7.5002 eV     [  0.2756 Ha]  -  Oscillator Strength:       0.0000
|          Transition Moments    X:    0.0000   Y:    0.0000   Z:    0.0000
|
|    4. Singlet:      7.6926 eV     [  0.2827 Ha]  -  Oscillator Strength:       0.3463
|          Transition Moments    X:    0.0000   Y:    0.0000   Z:    0.7207
...
```

All computed excitations are also written to an extra file `TDDFT_LR_Spectrum_Singlet.dat`

```
$ head -n 5 TDDFT_LR_Spectrum_Singlet.dat
            6.94239384535589                 0.09906927964673
            7.49811392742994                 0.00000000000000
            7.50015774774922                 0.00000000000000
            7.69255706787699                 0.34625377115159
            7.74808638563561                 0.00000000000000
```

This file can be used to plot a stick representation for the optical absorption spectrum of the molecule:

```
$ gnuplot
gnuplot> p [0:15] 'TDDFT_LR_Spectrum_Singlet.dat' w i
```

Note, that we restrict ourselves here to the lowest 15eV of the spectrum, since in general the Casida approach combined with PBE as ground-state functional and adiabatic LDA for the exchange correlation kernel performs best for low lying excitations.

Alternatively one can also broaden the stick spectrum. The FHI-aims package contains the utitlity `casida_broadener` which can be used for this task. The utility requires an input file with name `casida_parameters.dat`. The parameter file contrains a single line with numerical parameters. A typical file has the form

```
> cat casida_parameters.dat
0.1 1.0 0.2 0.0 0 1 200 800
```

From left to right the parameters have the following meaning: 1) Lorentz width, 2) Lorentz weigth, 3) Gauss width, 4) Gauss weight 5) temperature shift in percent, 6) intensity rescaling, 7) lower 8) upper limit of spectrum in nanometers.

Create a `casida_parameters.dat` file with the above content and run the utitlity `casida_broadener`. As output of the utility you will find the files `Broadened_Spectrum_eV.dat` and `Broadened_Spectrum_nm.dat` which contain boradened spectra.

Optional: H_2O Repeat the calculation for the water molecule H_2O, using the experimental geometry available at `http://cccbdb.nist.gov/`.

Exercise 7: Ground and excited state Born-Oppenheimer surfaces for C_2H_4

[*Total time for this exercise: 15 minutes. Total CPU time:* < 14 *minutes.*]

In this excercise we compute the ground and a few low-lying excited Born-Oppenheimer surfaces of C_2H_4 for different twisted geometries.

In order to generate geometries for different twisting angles and to run FHI-aims for these geometries, we have prepared a script that you can utilize for this task. Copy the script to your working directory

```
$ cp -a $HandsOnDFT/tutorials/tutorial5/reference/PartII/\
   exercise_7_excited_state_BO_surfaces/run-aims.sh .
```

and run the script. As output you will get a file `c2h4_bo_surfaces.dat` which contains the ground state and the lowest seven excited state surfaces. Plot the excited surfaces together with the ground state surface (columns 1 versus 2 gives the ground state surface, 1 versus 3 the first excited state surface, 1 versus 4 the second excited state surface and so on).

Comparison of algorithms for real-time evolution of wavepackets

[*Total time for this exercise: 50 minutes. Total CPU time:* ~ 30 *minutes.*]

The time evolution of a quantum mechanical state vector $|\Psi(t)\rangle$ can be expressed in terms of the time-ordered evolution operator

$$\hat{U}(t+\Delta t, t) = \hat{T} \exp\left(-i \int_t^{t+\Delta t} \hat{H}(\tau) d\tau\right), \tag{14}$$

where $\hat{H}(t)$ denotes the (possibly) time-dependent Hamiltonian of the system. With help of $\hat{U}(t+\Delta t, t)$ the state $|\Psi(t)\rangle$ can be propagated forward in time according to

$$|\Psi(t+\Delta t)\rangle = \hat{U}(t+\Delta t, t)|\Psi(t)\rangle. \tag{15}$$

Practical propagation schemes are therefore concerned with approximate representations of $\hat{U}(t+\Delta t, t)$, or to be more precise, with approximate ways to compute the action of $\hat{U}(t+\Delta t, t)$ on a state vector (with or without explicitly constructing the operator $\hat{U}(t+\Delta t, t)$).

Magnus expansion

The first complication that arises in approximating $\hat{U}(t+\Delta t, t)$ is the time-ordering operator $\hat{T}$ in Eq. (14). The Magnus series [11,12] provides an exact expression for the time-evolution operator Eq. (14) as a *time-unordered* exponential of so called Magnus operators Ω_j in the form

$$\hat{U}(t+\Delta t, t) = \exp\left(\hat{\Omega}\right) = \exp\left(\hat{\Omega}_1 + \hat{\Omega}_2 + \hat{\Omega}_3 + \cdots\right), \qquad \hat{\Omega} = \sum_{j=1}^{\infty} \hat{\Omega}_j, \tag{16}$$

where the $\hat{\Omega}_j$ are given in terms of time-integrals over nested commutators of the Hamiltonian at different points in time

$$\begin{aligned}
\hat{\Omega}_1 &= -i \int_t^{t+\Delta t} \hat{H}(\tau) d\tau \\
\hat{\Omega}_2 &= \int_t^{t+\Delta t} \int_t^{\tau_1} [\hat{H}(\tau_1), \hat{H}(\tau_2)] d\tau_2 d\tau_1 \\
&\vdots
\end{aligned} \tag{17}$$

The time-integrals can be evaluated numerically with e.g., a Gauss-Legendre quadrature. In the simplest case, which is accurate up to second order in the time-step Δt, one arrives at the exponential midpoint rule

$$\begin{aligned}
\hat{U}^{(2)}(t+\Delta t, t) &= \exp\left(\hat{\Omega}_1\right) + O(\Delta t^3) \\
\hat{\Omega}_1 &= -i\hat{H}(t+\Delta t/2)\Delta t + O(\Delta t^3).
\end{aligned} \tag{18}$$

Higher order truncations of the Magnus series can easily be taken into account but require the computation of additional commutators of the Hamiltonian at different points in time. For example, the fourth order Magnus propagator takes the form

$$\begin{aligned}
\hat{U}^{(4)}(t+\Delta t, t) &= \exp\left(\hat{\Omega}_1 + \Omega_2\right) + O(\Delta t^5) \\
\hat{\Omega}_1 &= -i(\hat{H}(\tau_1) + \hat{H}(\tau_2))\frac{\Delta t}{2} + O(\Delta t^5). \\
\hat{\Omega}_2 &= -i[\hat{H}(\tau_1), \hat{H}(\tau_2)]\frac{\sqrt{3}\Delta t^2}{12} + O(\Delta t^5). \\
\tau_{1,2} &= t + (\frac{1}{2} \pm \frac{\sqrt{3}}{6})\Delta t
\end{aligned} \tag{19}$$

Exponential of a matrix

When a discretized representation of the state vectors and the Hamiltonian is introduced (e.g. basis set, or finite difference approximation) the propagation step, Eq. (15) in combination with the Magnus expansion, Eq. (16), amounts to compute the action of a matrix exponential on a vector. In the following we discuss different ways to compute this operation.

- *Taylor expansion.* The simplest approach is to perform a Taylor expansion of the exponential

  $$\exp(\hat{\Omega}) = 1 + \hat{\Omega} + \hat{\Omega}^2/2 + \hat{\Omega}^3/6 + \hat{\Omega}^4/24 + O(\hat{\Omega}^5). \tag{20}$$

  Acting with this approximation on a state vector involves only matrix vector products which can be computed efficiently. From a stability analysis, it turns out that an expansion up to fourth order gives the best compromise of computational cost and stability.

- *Matrix diagonalization.* If we have all eigenvalues of the matrix representation of $\hat{\Omega}$ at hand, we can simply take the exponential of the eigenvalues and transform back to the original non-diagonal frame of the matrix. This approach gives the most accurate way to compute the matrix exponential. The price to pay is the large computational cost for the dense matrix diagonalization and the required matrix-matrix products.

- *Padé approximation.* The lowest order approximation of the exponential by rational functions takes the form

  $$\exp(\hat{\Omega}) \approx \frac{1 + \hat{\Omega}/2}{1 - \hat{\Omega}/2}. \tag{21}$$

  This leaves us with an operator (or in discretized form with a matrix) in the denominator. However, since we are only interested in the action of the approximated exponential on a state vector this can be written as

  $$(1 - \hat{\Omega}/2)|\Psi(t + \Delta t)\rangle = (1 + \hat{\Omega}/2)|\Psi(t)\rangle. \tag{22}$$

  For a given point in time t, we can explicitly construct the right hand side of Eq. (22). The state vector at time $t + \Delta t$ can then be found by solving a linear system of equations. This scheme is also known as Crank Nicholson or Cayley approximation.

Standard ODE solver

In case we only consider a discretization of the state vectors and the Hamiltonian, we can regard the time-dependent Schrödinger equation as set of coupled first order ordinary differential equations in time

$$\frac{d}{dt}\Psi_p(t) = \sum_q -iH_{p,q}(t)\Psi_q(t). \tag{23}$$

For each given index p this corresponds to one ordinary differential equation (ODE) which is coupled to the other ODEs through the matrix elements $H_{p,q}(t)$ of the Hamiltonian. In this form standard ODE solver can be employed. Particularly simple ODE time-stepping schemes are Runge-Kutta methods. A frequently employed discretization is the Runge-Kutta method of fourth order

$$\begin{align} y' &= f(t, y), \qquad y(t_0) = y_0 \tag{24}\\ k_1 &= hf(t_n, y_n) \tag{25}\\ k_2 &= hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \tag{26}\\ k_3 &= hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \tag{27}\\ k_4 &= hf(t_n + h, y_n + k_3) \tag{28}\\ y_{n+1} &= y_n + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4] \tag{29}\\ t_{n+1} &= t_n + h \tag{30} \end{align}$$

which in terms of our TDSE notation can be written as

$$\Phi_p^{(1)} = \sum_q -i\Delta t H_{p,q}(t_n)\Psi_q^{(n)} \tag{31}$$

$$\Phi_p^{(2)} = \sum_q -i\Delta t H_{p,q}(t_n + \frac{1}{2}\Delta t)[\Psi_q^{(n)} + \frac{1}{2}\Phi_q^{(1)}] \tag{32}$$

$$\Phi_p^{(3)} = \sum_q -i\Delta t H_{p,q}(t_n + \frac{1}{2}\Delta t)[\Psi_q^{(n)} + \frac{1}{2}\Phi_q^{(2)}] \tag{33}$$

$$\Phi_p^{(4)} = \sum_q -i\Delta t H_{p,q}(t_n + \Delta t)[\Psi_q^{(n)} + \Phi_q^{(3)}] \tag{34}$$

$$\Psi_p^{(n+1)} = \Psi_p^{(n)} + \frac{1}{6}[\Phi_p^{(1)} + 2\Phi_p^{(2)} + 2\Phi_p^{(3)} + \Phi_p^{(4)}] \tag{35}$$

$$t_{n+1} = t_n + \Delta t \tag{36}$$

Although such a Runge-Kutta algorithm is easy to implement it does not numerically exhibit the time-reversal symmetry which is a property of the exact solution of the TDSE (provided magnetic fields are absent).

Autocorrelation function

One possibility to compute the spectrum of the Hamiltonian $\hat{H}$ via real-time propagation relies on the autocorrelation function of the propagated state vector. The free time-evolution (without external field) of a state vector is given as

$$|\Psi(t)\rangle = \sum_q a_q|\Psi_q\rangle \exp(-iE_q t), \tag{37}$$

where $|\Psi_q\rangle$ and E_q are the solutions of the static Schrödinger equation

$$\hat{H}|\Psi_q\rangle = E_q|\Psi_q\rangle \tag{38}$$

and a_q are time-independent expansion coefficients. Next, let us consider the time-dependent correlation function

$$P(t) = \langle\Psi(t=0)|\Psi(t)\rangle. \tag{39}$$

Using Eq. (37) this can be written as

$$P(t) = \sum_q |a_q|^2 \exp(-iE_q t). \tag{40}$$

By computing the Fourier Transform of the correlation function $P(t)$, we arrive at a spectrum which has peaks at the eigenenergies of the system

$$P(E) = \sum_q |a_q|^2 \delta(E - E_q). \tag{41}$$

The peaks in the spectrum appear infinitely sharp since we assumed a infinite-length time-record of the autocorrelation function $P(t)$. If only a finite-length time-record is available, then a broadening of the peaks occurs. This broadening is solely due to the finite propagation time and should not be confused with the natural line width of electronic levels.

Exercise 8: comparison of different real-time propagation algorithms

In this exercise we compare the efficiency and accuracy of the different propagation schemes that have been introduced in the previous section.

In the folder

```
$HandsOnDFT/tutorials/tutorial5/reference/PartII/exercise_8_propagation_algorithms
```

you can find a few Matlab/Octave [5] scripts which contain a simple implementation of the discussed algorithms. Copy these files to your current working directory

```
$ cp -a $HandsOnDFT/tutorials/tutorial5/reference/PartII/exercise_8_propagation_algorithms/*.m .
$ ls -1 *.m
autocorrelation_spectrum.m
double_well_hamiltonian.m
matrix_exponential_taylor.m
oscillator_hamiltonian.m
propagation_algorithms.m
real_space_grid.m
td_hamiltonian.m
```

For simplicity, the implementation of our example is using a one-dimensional finite-difference representation of Hamiltonians and state vectors. The main program is contained in the file propagation_algorithms.m. The files double_well_hamiltonian.m and oscillator_hamiltonian.m contain different definitions of the system Hamiltonian (asymmetric double well and harmonic oscillator), the file real_space_grid.m contains the definition of the employed real-space grid and td_hamiltonian.m allows to construct explicitly time-dependent Hamiltonians. Take a look at all files to familiarize yourself with the propagation algorithms.

To start the program you have to execute

```
$ octave propagation_algorithms.m
```

or alternatively you can use the octave shell

```
$ octave
octave:1> propagation_algorithms
```

The numerical settings can be adapted by editing the script propagation_algorithms.m. In the file you can find the parameters for the grid spacing, number of time steps, etc.

```
max_time_steps = 4096; % maximum number of time steps
dt = 0.024;            % time step
nx = 100;              % number of grid points
dx = 0.16;             % grid spacing
```

You can switch between different propagation schemes by adjusting the flags for the propagators

```
% propagation algorithms
rti_cn             = true;   % Crank-Nicholson/Cayley propagator
rti_rk4            = false;  % Runge-Kutta 4th order
rti_taylor         = false;  % Taylor expansion of exponential
rti_expm           = false;  % Exact matrix expoential
rti_etrs           = false;  % Enforced time-reversal symmetric propagator
rti_magnus2        = false;  % Second order Magnus expansion with exact matrix expoential
rti_magnus2_taylor = false;  % Second order Magnus expansion with Taylor expansion of exponential
```

[5] Matlab is a powerful numerical computing environment. The package allows for easy matrix and vector manipulations and provides extensive plotting routines for numerical data. GNU Octave is an open source implementation of the Matlab computing environment and programming language.

Only one `rti_*` variable should be set to true for a given run.

In the following steps we compare two different propagation schemes (Crank-Nicholson/Cayley and Runge-Kutta 4th order). The initial Hamiltonian corresponds to a finite-difference discretization of a one-dimensional harmonic oscillator.

1. Run the script `propagation_algorithms.m` with the default parameters. Once the run finished, inspect the generated PNG files by typing in the shell

   ```
   $ qiv *.png
   ```

   Press the space bar to switch to the next image. You will see among others the spectrum, Eq. (41), of the autocorrelation function, Eq. (39), the lowest four eigenstates of the Hamiltonian plotted together with the potential, also the sparsity pattern of the Hamiltonian matrix (zero elements are white, non-zero elements are blue) and the sparsity pattern of the exponential of the Hamiltonian.

2. Compute the autocorrelation function for a different initial condition. For example, change the parameter `xmin` in line 106 of the file `propagation_algorithms.m` from 1.0 to a value of 4.0. How many spectral lines can be resolved in both cases? What is causing the difference?

3. Following the formulas of Eqs. (31) - (36), try to add the fourth-order Runge-Kutta time-stepping to the provided file `propagation_algorithms.m`. As a hint you can find Eq. (31) and Eq. (32) as example in the script

   ```
       rk1 = -sqrt(-1) * dt * Hm_t     * (psi);
       rk2 = -sqrt(-1) * dt * Hm_t_dt2 * (psi + rk1/2);
   % >>>>>>>> Add missing lines
   %   rk3 =
   %   rk4 =
   % <<<<<<<< Add missing lines
       psi = psi + (rk1 + 2*rk2 + 2*rk3 + rk4)/6;
   ```

   Run the Runge-Kutta time-stepping by adjusting the corresponding flags:

   ```
   % propagation algorithms
   rti_cn             = false;   % Crank-Nicholson/Cayley propagator
   rti_rk4            = true;    % Runge-Kutta 4th order
   ```

   Can you still reproduce the same spectrum for the autocorrelation function? Which algorithm runs faster, Runge-Kutta, or Crank-Nicholson/Cayley? Is the norm of the wavefunction still preserved when you increase the time-step for Runge-Kutta from `dt = 0.027` to `dt = 0.028`? What can you say about the stability of the Runge-Kutta and Crank-Nicholson/Cayley schemes for larger time steps with `dt > 0.027`?

4. By changing the parameters `dt` for the time-step and `nx` for the grid-spacing, try to investigate for the Crank-Nicholson/Cayley and the Runge-Kutta propagation scheme the following questions:

   - Which method allows for the largest time step?
   - Which method allows for the fastest time stepping?
   - How is the grid spacing related to the time step?

   The stability of a given propagator can be monitored by plotting the energy and norm of the state as function of time. If the norm is not approximately unity (e.g. starts to blow up and eventually shows `NaN` (not a number)), the numerical propagation is not usable anymore.

5. Optional: If time permits also compare to other propagation schemes, e.g. the exact matrix exponential (Option `rti_expm`).

6. Enable the frame output by setting

   ```
   td_frame_output = true;       % output frames for wave packet movie
   ```

If this is enabled, the script `propagation_algorithms.m` also generates a movie of the computed wavepacket propagation. Also set

```
tdmod = 8;
xmin = 1.0;
```

and run the script. You can display the time-evolving wavefunction by typing in the shell e.g.

```
$ mplayer wave_packet.avi
```

Modify the initial condition of the time-propagation by changing the center and the "squeezing" parameter of the Gaussian initial state.

```
% initial state ((shifted and squeezed) Gaussian at minimum of potential)
% coherent state (xmin != 0.0, squeeze = 1.0)
% squeezed state (xmin != 0.0, squeeze != 1.0)
xmin = 0.5;
squeeze = 2.0;
psi = sqrt(dx)*(mass*(omega/squeeze) / pi)^(1/4)*exp(-0.5*mass*(omega/squeeze)*(x+xmin).^2);
```

How does the observed wave-packet propagation compare to a classical oscillator?

7. Switch to a Hamiltonian for an asymmetric double well by commenting the line containing `oscillator_hamiltonian` and uncommenting the line containing `double_well_hamiltonian`

   ```
   [Hm, vpot] = double_well_hamiltonian(nx, dx, x, B, w0, beta);
   %[Hm, vpot] = oscillator_hamiltonian(nx, dx, x, mass, omega);
   ```

   Run the script and inspect again the spectrum of the autocorrelation function and the movie of the wavepacket propagation.

Exercise 9: Wavepacket dynamics on coupled Born-Oppenheimer surfaces

In the last excercise of the tutorial we propagate nuclear wavepackets on coupled Born-Oppenheimer surfaces. To that end we use the ground and first-excited state potential energy surfaces from the TDDFT/Casida runs in exercise 7 and combine this with the propagation algorithms that we introduced in exercise 8.
To start the excercise, create a new folder and copy the script

```
    $HandsOnDFT/tutorials/tutorial5/reference/PartII/\
      exercise_9_PES_propagation/PES_propagation.m
```

to your working directory. Also copy the file `c2h4_bo_surfaces.dat` with the Born-Oppenheimer surfaces from exercise 7 to your current folder. You can then launch the wavepacket propagation by typing in the shell

```
$ octave
octave:1> PES_propagation
```

In total the script will perform 3000 time steps. As output the script generates snapshots of the moving wavepacket at different points in time. From these snapshots you can generate a movie of the time propagation by executing

```
$ ffmpeg -sameq -y -r 20 -i frame%05d.png bo_surfaces_propagation.mp4
```

Finally, in order to watch the movie you can use

```
$ mplayer bo_surfaces_propagation.mp4
```

The initial state for the dynamics is the ground-state wavefunction of the ground-state surface, but placed in the energy landscape of the first excited-state surface. This corresponds to a vertical Franck-Condon transition. As you can see in the dynamics, the wavepacket starts to split and also coupling between the ground-state and first-excited state surfaces is visible.

References

[1] Chong, Gritsenko and Baerends, *J. Chem. Phys.* **116**, 1760 (2002);
[2] G. Bieri and L. Åsbrink, J. of Electron Spectrosc. and Rel. Phenom. **20**, 149 (1980);
[3] Bagus, *Phys. Rev.* **139**, A619 (1965);
[4] Hedin, *Phys. Rev.* **139**, A796 (1965);
[5] Aryasetiawan and Gunnarsson, *Rep. Prog. Phys.* **61**, 237 (1998);
[6] R.D. Mattuck, A guide to Feynman diagrams in the Many-Body problem, Dover Books;
[7] F. Caruso, P. Rinke, X. Ren, A. Rubio, and M. Scheffler, Phys. Rev. B, accepted, (2013).
[8] N. E. Dahlen, R. Van Leeuwen, and U. Von Barth, Phys. Rev. A **73**, 012511 (2006).
[9] M. E. Casida, *Time-dependent density functional response theory for molecules* in Recent Advances in Density Functional Methods, Ed. D. E. Chong, World Scientific, (1995).
[10] Miguel A. L. Marques, Micael J. T. Oliveira, and Tobias Burnus, Libxc: a library of exchange and correlation functionals for density functional theory, Comput. Phys. Commun. **183**, 2272 (2012).
[11] W. Magnus, Comm. Pure Appl. Math. 7 (1954) 649;
[12] A. Castro et. al., J. Chem. Phys. 121, 3425 (2004);

Ionization energies of C_2H_4 and H_2O

C_2H_4	Experiment	DFT (PBE)	Hartree-Fock	Δ-SCF	G_0W_0@PBE	G_0W_0@HF	scGW
HOMO	10.68 eV						
HOMO-1	12.8 eV			***			
HOMO-2	14.8 eV			***			
HOMO-3	16.0 eV			***			

Optional:

H_2O	Experiment	DFT (PBE)	Hartree-Fock	Δ-SCF	G_0W_0@PBE	G_0W_0@HF	scGW
HOMO	12.615 eV						
HOMO-1	14.729 eV			***			
HOMO-2	18.550 eV			***			