Deckblatt: Revision 1.1 + Sub-project 1.2

Contents

1 What changed regarding our first submission **2**
1.1 Modularity 2
1.2 Variables 2
1.3 Tests 3
1.4 Convergence with rel() function xxxxxxxchange title 5

2 How the code works for the second sub-project **5**

3 Workflow **5**

4 Implementing conjugate-gradient and power method **5**

5 Testing **5**

6 Convergence of eigenvalues/eigenvectors **5**

7 Appendix **6**

1 What changed regarding our first submission

1.1 Modularity

As a first step, we improved our file management by implementing a modularity of files, dividing our prior one python file into seven (and of course more files for the second sub-project). The content of these modules did not change if not stated otherwise in the sections Variables, Tests or Convergence.
The new file system is listed below:
- "variables.py": defining variables and few general functions
- "hamiltonian.py": defining potential and hamiltonian
- "test_hamiltonian.py": defining the tests for the hamiltonian and also running them for multiple N and dimensions
- "integrators.py": defining both the integrators
- "test_integrators.py": defining the tests for the integrators and also running them for multiple N and dimensions
- "test_integr_converg.py": testing the dependence of the integrators on M or rather τ
- "animation.py": defining functions for animation and animating

1.2 Variables

We then changed/deleted all our old variables depending on other parameters to be now only defined by dimensionless values, without changing their value in regards to our first submission. In our case of modularity where we work with the same variables in multiple python files it is also practical to add an extra python file for storing our constant variables (and some basic functions used in multiple other files). We can then import these global variables by "from variables import *".
This change worked with most of the prior functions and tests except when needing to change certain

variables of a function in a separate file from where the function is defined. Explicitly, what changed:
1. For the new hamiltonian tests (see 1.3), we had to change our definition of the potential to take N from the size of the given array instead of using the global N.
2. A second change regarding the hamiltonian file was done. We defined potential_variable and hamilton_variable which are the same as our old potential and hamiltonian but take μ and ϵ as an input for the convergence of eigenvalues/-vectors in the second sub-project.
3. Similarly, but for the first sub-project, for testing the integrators in dependence of M or rather τ, we had to add an additional input for τ into the integrators (and therefore the tests). Therefore, the dependence could be evaluated without having to define the integrators in the same file as the tests, which was done in our first submission.

1.3 Tests

The tests we included in the sub-project 1 were in general not modified for this revision. We now ran these tests of the hamiltonian and integrators for multiple dimensions and N for proper testing. Here we take focus on $N = 5$, 10, 15 and 20; because the relevance of boundary conditions and resulting errors decrease for large N. The exact output for running the testing files is shown in the following figures for each of the N for 4 dimensions (hamiltonian) and 3 dimensions (integrators, due to long run-time). The tests are including "randomly" generated arrays. Therefore these tests run 10 times each and depending on the criteria the maximum error of each test is saved.
Also notice that the output shows N as a float, it is an integer nonetheless.

```
Testing linearity of the hamiltonian 10 times. Maximum error:
     N                     1D                     2D                     3D                     4D
   5.0  1.1546319456101628e-14   3.260957681782951e-14  5.123796534383003e-14    6.07087979554246e-14
  10.0  1.5888218580782548e-14  3.1776437161565096e-14  5.955433342649105e-14    8.728132538933693e-14
  15.0   1.551136925266137e-14  4.0194366942304644e-14  6.204547701064548e-14    9.237055564881302e-14
  20.0  1.5888218580782548e-14   3.552713678800501e-14  6.355287432313019e-14  1.4228607195221903e-13

Testing hermiticity of the hamiltonian 10 times. Maximum error:
     N                     1D                    2D                      3D                     4D
   5.0   2.864289338439558e-14  8.987733679556355e-14   2.428351918216984e-13  1.8269655220124374e-12
  10.0  1.5888218580782548e-14  1.156679772130055e-13  1.8749713606747085e-12    2.910738295359853e-11
  15.0  3.1776437161565096e-14  4.602649920770331e-13   7.279509460923746e-12  1.1647215137477994e-10
  20.0   3.552713678800501e-14  4.856703836433968e-13  1.4551915228366852e-11    6.984942994333792e-10

Testing positivity of the hamiltonian 10 times. Number of times hamiltonian was negative:
     N 1D 2D 3D 4D
   5.0  0  0  0  0
  10.0  0  0  0  0
  15.0  0  0  0  0
  20.0  0  0  0  0

Testing eigenvectors of the kinetic hamiltonian 10 times. Maximum error:
     N                     1D                     2D                      3D                     4D
   5.0  1.2434497875801753e-14   6.394884621840902e-14    3.377881604366369e-13  1.2477715126847702e-12
  10.0  1.3756211832640111e-13   2.602423395528474e-13    4.990013697488234e-13   8.676771314227237e-13
  15.0  1.8898569046946333e-13  1.2216796498472902e-12   1.5157884509969167e-12   1.879116880688524e-12
  20.0  1.9657657907861573e-13    7.06280573930772e-13    1.305019366642739e-12  3.4225229075923987e-12
```

Figure 1: Output of our test functions for the hamiltonian for multiple N and D; 10 iterations.

One can see that the errors for testing the linearity and eigenvectors are in the range of 10^{-14} to 10^{-13}, where for higher dimensions the error increases. It also seems that for higher N the errors also increase. But in some cases this could also be due to the "random" generation. We observe the same dependence for the hermiticity with errors up to 10^{-10} and eigenvectors with error up to 10^{-12}. The positivity of the hamiltonian was correct all the iterations.

Now our integrator tests.

```
Testing unitarity of the Second-Order integrator 10 times. Maximum error:
    N                       1D                       2D                      3D
  5.0  3.9904181437844244e-05  0.00013276611390056203   0.0003624607222638687
 10.0   1.386199613206962e-05   8.927968384719875e-05  0.00033131230876581874
 15.0  1.2751883163408806e-05    9.56638492204398e-05   0.0003168524692618302
 20.0  2.3743908606377317e-05  0.00010123774259285234  0.00031313793351417374

Testing unitarity of the Strang-Splitting integrator 10 times. Maximum error:
    N                       1D                     2D                     3D
  5.0  3.1086244689504383e-15  4.884981308350689e-15  4.884981308350689e-15
 10.0  4.6629367034256575e-15  5.773159728050814e-15  4.440892098500626e-15
 15.0   5.218048215738236e-15  6.217248937900877e-15  6.106226635438361e-15
 20.0   2.886579864025407e-15  5.773159728050814e-15  7.771561172376096e-15

Testing linearity of the Second-Order integrator 10 times. Maximum error:
    N                       1D                     2D                     3D
  5.0  3.1401849173675503e-16  4.577566798522237e-16  4.965068306494546e-16
 10.0    4.47545209131181e-16  4.965068306494546e-16  9.036560719766055e-16
 15.0   4.965068306494546e-16   9.09180472805593e-16   8.95090418262362e-16
 20.0    4.47545209131181e-16  9.036560719766055e-16  9.694605782913356e-16

Testing linearity of the Strang-Splitting integrator 10 times. Maximum error:
    N                      1D                      2D                      3D
  5.0  6.280369834735101e-16    8.95090418262362e-16  1.3732700395566711e-15
 10.0  7.021666937153402e-16  1.0190492755571723e-15   1.336885555457667e-15
 15.0   8.95090418262362e-16   9.992007221626409e-16  1.4217791915866692e-15
 20.0  9.155133597044475e-16  1.3608726004012153e-15  1.4217791915866692e-15

Testing energy conservation of the Second-Order integrator 10 times. Maximum error:
    N                       1D                     2D                    3D
  5.0  0.00016577215673052592  0.0009314073064423667  0.005688503833425784
 10.0    7.09790981012759e-05  0.0010863070875046787  0.005532682767821484
 15.0  0.00010139364897909786  0.0012747319324795114  0.005252730649537796
 20.0  0.00012092889763337666  0.0012254080048776927  0.005115145558249878

Testing energy conservation of the Strang-Splitting integrator 10 times. Maximum error:
    N                   1D                  2D                  3D
  5.0  0.5033770851718984  4.5220841571248656    8.13412230086044
 10.0  1.2348800258027488  3.5423718734041785  10.882246287871567
 15.0  0.3926619525378072  3.3853661925469822   8.593889078387669
 20.0  0.5061939239646911  2.4737736743123904    6.36787533429078
```

Figure 2: Output of our test functions for the integrators for multiple N and D; 10 iterations.

The errors of unitarity for the strang-splitting integrator and linearity for both of the integrators are in the range of $10^{-16} - 10^{-15}$, which is even smaller than for the hamiltonian tests and checks with the expectation. The unitarity errors for the second-order integrator are far larger even in 1D with 10^{-5} and rising for higher dimensions. Even greater errors can be seen for the energy conservation tests. This is expected as energy conservation is not given for these integrators. The important detail however, is that no significant lowering of errors for higher N is seen, which is most likely due $N = 20$ still being relatively small and using randomly generated arrays, which change the error maxima.

Below, all the tests were run for one iteration for $N = 5$ and $D = 9$ as an additional testing. One can see the same behaviour as above, just exaggerated.
With all these evaluated results we can safely conclude that our tests run (correctly) in more dimensions than $D = 1$, even for small N.

```
linearity:  1.9895196601282805e-13
hermiticity:  2.1827872842550278e-11
positivity:  0
eigenvectors:  7.503546602517921e-13
```

Figure 3: Output of our test functions for the hamiltonian for $N = 5$ and $D = 9$; 1 iteration.

```
unitarity so:  0.00011173011079268491
unitarity ss:  2.220446049250313e-16
linearity so:  6.753223014464259e-16
linearity ss:  1.1957467920563633e-15
energy conserv. so:  0.043701583997005855
energy conserv. ss:  0.7337248193550607
```

Figure 4: Output of our test functions for the integrators for $N = 5$ and $D = 9$; 1 iteration.

1.4 Convergence with rel() function xxxxxxxchange title

- Flo H

2 How the code works for the second sub-project

3 Workflow

4 Implementing conjugate-gradient and power method

5 Testing

!!! describe this time with detail and test all possible configurations!!!

6 Convergence of eigenvalues/eigenvectors

7 Appendix