GOScalc

Introduction and credits

This is a program to calculate the total generalized oscillator strength for a certain (sub-)orbital. For specific information refer to [1] (German). The project was taken over from Dr. Stephan Müller. The additional program Wavegen is based on [2] and was provided by Prof. Krüger, who modified it together with M.Frigge [3].
The Hankel transformation in hankel_trafo.cc is an adapted version of NumSBT (aanz_v2), which was written in Fortran90 and published as [4].

Installation and compiling (Linux systems):

Libraries used: Armadillo (requiring standard BLAS+LAPACK or OpenBLAS), FFTW, Boost, WignerSymbols. WignerSymbols can be found here
Make sure you have cmake, make and a fortran and c++ compiler installed (gfortran, gcc).

Create directory build (if it doesn't exist already) and cd into it:

```
mkdir build/
cd build/
```

Compile with cmake and make:

```
cmake ..
 make
```

Compile wavegen_mod with:

```
gfortran -std=legacy -o wavegen_mod wavegen_mod.f
```

Usage

- create the wavegen.dat file:
    - The config file for wavegen should be named wavegen.dat and be placed in the same directory as wavegen.
    - It tells the program which exchange-correlation (XC) functional to use, the atomic number (Z) of the simulated atom and its electron configuration.
        - Options for the XC functional are LDA (Local-Density Approximation) and GGA (Generalized Gradient Approximation). LDA is recommended.
    - The electron configuration is given in rows, where each row contains the principal quantum number (n), the azimuthal quantum number (l) and the occupation number (ON) separated by spin direction (ONup and ONdown).
    - Conceptually, the wavegen.dat then looks like this:

      ```
      XC-functional
      Z
      n l ONup ONdown
      n l ONup ONdown
      ..............
      n l Onup Ondown
      ```

    - With the quantum numbers n,l and the number of electrons with spin up or down for that

subshell. Empty shells should not be listed.

- See Example Files for examples of wavegen.dat

- Example files are provided for copper, silicon, carbon and lead.
  - These need to be renamed to wavegen.dat for usage.
  - Notice: The spin is not maximized correctly as according to Hund's second rule.
    - This is because goscalc doesn't include spin effects. So instead spin is equally distributed; with the excess electron for odd numbers of electrons put as spin down because later the file waveup.dat (containing electrons with spin up) is used (this reduces the effect of the imbalance). Electron configurations can be looked up here.
- execute wavegen in the same directory

./wavegen

- ignore the warning about floating point exceptions (doesn't seem to matter)
- place the waveup.dat file in the same directory as the config.json and the goscalc executable

- fill in config.json with the desired parameters and the output directory name
  - the config.json looks like this:

```
{
  "dft_filename": "waveup.dat",
  "output_dir_name": "C",
  "n_bound": 1,
  "l_bound": 0,
  "max_considered_lfree": 15,
  "energy_free_start": 10.5,
  "energy_free_steps": 8,
  "energy_free_increase":20,
  "max_kvalue_Ang": 70
}
```

  - These values denote:
    - dft_filename is the name of the file given by wavegen containing the bound electron wave functions
    - output_dir_name is the name the output files will be written to
    - n_bound, l_bound denote the subshell for which the GOS is to be computed
    - max_considered_lfree is the maximum angular quantum number l to be considered for the ejected electron
    - energy_free_start is the lowest energy loss for which the GOS will be computed
    - energy_free_steps the number of energy loss steps for which the GOS will be computed
    - energy_free_increase the size of each of these steps
    - max_kvalue_Ang up to which wavenumber k_{N-1} value the GOS will be computed. This is used in conjuction with the size of the real space lattice (r_0 and r_{N-1}) given in the wavegen output file to determine the minimum wavenumber k_0 = k_{N-1} * r_0 / r_{N-1}
  - See Example Files for examples of config.json
  - execute goscalc

./goscalc

- alternatively you can pass the path to the config file as a command line argument

./goscalc /path/to/config.json

- The mesh goscalc uses is the one given by wavegen (and the reciprocal lattice is inferred in combination with max_kvalue_Ang in config.json, see above). To change the number of mesh points or the mesh parameter you have to edit the parameters mmax and rmax, respectively, in wavegen_mod.f.

Output:

The output directory includes a copy of the config file, the command line log, the k values in k.dat, the corresponding generalized oscillator strengths in gos.dat for the energy losses, which result from the desired free energies saved in free_energies.dat.

Example files:

element_configs contains some example configuration files to use with wavegen and goscalc for different elements XY. The directories XY_cfg contain the config files for wavegen (wavegen.dat) and goscalc (config.json) as well as the resulting wavefunctions calculated by wavegen (waveup.dat). You can check your results, by comparing them to the output files given in element_configs/XY

Restrictions:

- GOS for ions can not be calculated, because their atomic potential does not fall off to zero within the mesh given by wavegen (or at all, technically), but contwace.c requires this.
- the number of mesh points is hard coded into wavegen. It can be changed by changing mmax. It should probably be a power of 2, if not just for efficiency reasons.

Known Issues:

- When compared to the GOS tables used by Gatan's EELS Analysis (2.3.2) there is a significant qualitative difference in the GOSs, while the overall shape is very similar. The difference appears to be stronger for higher l, though this hasn't been tested rigorously.
- Mesh sizes of mmax=2^14 or higher lead to issues in wavegen.

Bibliography

[1] Leonhard Segger, Berechnung generalisierter Oszillatorenstärken für die Quantifizierung von EEL-Spektren, Bachelorarbeit, WWU-Münster 2019 (soon available at https://www.uni-muenster.de/Physik.PI/Kohl/pub.html)
[2] D. R. Hamann, Phys.Rev. B 40 (1989), 2980 https://journals.aps.org/prb/abstract/10.1103/PhysRevB.40.2980
[3] Frigge, Kohl, Krüger, Microscopy Conference 2011 (Kiel), IM5.P174, Calculation of relativistic differential cross-sections for use in microanalysis. Abstract available at https://www.uni-muenster.de/imperia/md/content/physik_pi/kohl/mc2011/im5_p175.pdf
[4] P. Koval, J. D. Talman, Comp. Phys. Comm 180-2 (2009), 332-338 https://www.sciencedirect.com/science/article/pii/S0010465508003329
v2: https://data.mendeley.com/datasets/y294ttxyw4/1
v3: https://data.mendeley.com/datasets/m3fc83rytv/1