

Program **fdacc**

fdacc (Finite Difference Anharmonic Constants Calculator): Parallel and efficient anharmonic constants calculator to compute the necessary terms for the SCTST reaction rates with the `paradensum`, `parsctst` and `bdens` programs.

Program Authors: G. Mandelli, C. Aieta, L. Corneo, M. Ceotto
Contact: Michele Ceotto: michele.ceotto@unimi.it

fdacc is a hybrid Bash/Fortran code which utilizes a parallel implementation strategy to calculate anharmonic constants for minimum and transition state geometry calculations. It is designed to couple with any electronic structure software capable of Single Point Energy (SPE) calculations. The **fdacc** software is currently in beta release and it is interfaced with the widely used Gaussian16 ab initio code. However, due to its modular design, it can easily be interfaced with any other ab initio code¹.

This guide for the use of the **fdacc** program is structured as follows: we start with the program installation, then we explain the possible options to run the program, and finally, we give some practical guidelines and examples.

Program Installation

The following assumptions are necessary to run the program in the current version:

- The **fdacc** program is designed to work with the SLURM workload manager on Linux distributions. (Adjustments can be made to make it work on other systems). The use of a workload manager is necessary as it will be required to manage a large amount of parallel jobs.
- If the Gaussian program is being used, the compatible version is g16; no tests have been made on other versions. To use other Gaussian versions or to change the way the ab initio program is called, enter the **fdacc** folder and open the `prog_core_2` file. On line 15, change the variable `g_key` with the command to launch the Gaussian program on your system.

Before installing the program, place the fdacc folder in the desired and accessible location. Then, enter the fdacc folder and open the Makefile. Next, change the following lines to match your system specifications:

Line 8: progmod=module load progfile\chem-phys gl6. This variable must be set in case a module has to be loaded on a specific system environment to run the ab initio program. Change the variable value to the module load string you use in your system. Be careful to use the "\" notation instead of the slash symbol "/".

Line 9: cores=12. This variable sets the number of parallel cores per node for the geometry optimization. The same parallel cores will be used for the Hessian calculation step if the user chooses to compute it with the ab initio code. The reason for this choice is to take advantage of a possible ab initio code routine to get analytic gradients and/or Hessians. The availability of such routines depends on the specific ab initio code.

If you do not need to specify a given variable or load a module, leave the variables void in the Makefile.

To proceed with the program installation, you need to enter the fdacc folder and type:

```
./setup -install
```

If you change something in the Fortran subroutines or if you change the location of the folder containing the fdacc program, you need to uninstall it first using the following command:

```
./setup -remove
```

After the un/installation, you need to source the .bashrc file in your home directory with the command

```
source ~/.bashrc
```

or refresh the terminal. Now the fdacc command can run from any location in the system.

Program Execution

Once the program has been installed, it can be called from any directory running the following command:

`fdacc`

Additional options are possible:

- `fdacc -help`, prints a help message with essential instructions to run the code
- `fdacc -relaunch`, will restart the calculation from the first step (optimization or harmonic analysis)
- `fdacc -clean`, will clean the scratch folder where the calculations are saved. This scratch space is located in the `fdacc` folder.

To run the code, first create a working directory. Each anharmonicity matrix computation requires a distinct working directory. Then, a file **input.dat** file must be placed in the launching directory. Note the name of the file is fixed.

fdacc Input Data File format (input.dat)

Line 1: Calculation_type

R calculation of the anharmonicity matrix for a stable structure, corresponding to a minimum.

TS calculation of the anharmonicity matrix for a transition state structure, corresponding to a saddle point.

Line 2: program_name (desired ab initio program used)
=gaussian uses the Gaussian ab initio code.

Line 3: n_proc Number of cores per node in your machine

Line 4: wall_time Wall time for optimization and Hessian calculation (if carried out by the ab initio code). Time format: hh:mm:ss

Line 5: fin_diff

yes = Compute the Hessian with finite difference approach. Specifically, the code uses a centered second order approximation with a standard step size of 0.001.

no = Compute the Hessian analytically with the ab initio code.

Line 6: Electronic structure method used for the optimization step.

Line 7: Electronic structure method used for the Hessian calculation.

Line 8: Electronic structure method used for the anharmonic constants calculations.

Line 9: Basis set used in the optimization step.

Line 10: Basis set used in the Hessian calculation.

Line 11: Basis set used in the anharmonic constants calculations.

NOTE: the syntax to specify the electronic structure method and basis set are the ones used in the ab initio code.

Line 12: Launch command suitable for your SLURM scheduler, common commands are qsub and sbatch.

Line 13: gp Group launch parameter. Maximum number of simultaneously allocated nodes. The default ($gp = 0$) allow the code to use all available nodes in the system. It is important to specify the gp parameter when a significant amount of I/O disk operations are carried out. Furthermore, the user can set the gp variable to limit the number of nodes dedicated to the fdacc calculations.

Line 14: exclude_modes List of the normal mode indexes to be excluded from the anharmonic constant calculations. The modes are ordered in ascending order according to the value of the frequency. For TS structures the imaginary frequency mode is always listed first. The list of ordered harmonic frequencies is printed in the fdacc.out file. The order can be altered by specifying the ORDER.dat file (see below).

= 0 (default value) all the modes are considered for the anharmonic constant calculations.

=1 2 modes 1 and 2 are excluded from the anharmonic constant calculations.

Line 14: Job_name a string identifying the specific anharmonic matrix calculation process. For each Job_name a scratch folder will be created into the fdacc directory to hold the intermediate files for each calculation step.

Line 15: spe_time Wall-time for a single point energy calculation on a single core expressed in minutes.

Line 16: “cccccccccccccccc” motif (not to be changed)

Line 17+nlines: SLURM launching script template **this is system dependent.**

Note: The example folder in the fdacc directory provides a commented launching script template (input_launch_directives.sh). Please, delete the comments when copying the template.

Line 17+nlines+1: “cccccccccccccccc” motif (not to be changed)

Line 17+nlines+2: Charge, spin

charge	Total molecular charge
spin	Molecular spin multiplicity

Line 17+nlines+3: Input geometry in the xyz format: Element symbol, x, y, and z coordinates (Ångström), separated by spaces or tabs.

Standard fdacc execution (with option fin_diff = no)

The *fdacc* command is run several times to complete the anharmonicity matrix calculation. Thanks to this feature, the user can control intermediate results at each step. We describe below a standard procedure that comprises five steps:

1. After the preparation of a working directory and an input file (see the previous section), the first execution of the *fdacc* command from the working directory will launch the geometry optimization. Fdacc uses the geometry in the input.dat file as a starting guess. The level of electronic structure theory is also specified in the input.dat file. The fdacc code takes care of the correct setting of geometry optimization thresholds needed for the subsequent calculation of the anharmonicity matrix.

2. Once the geometry optimization is done, a second execution of the *fdacc* rotates the geometry into the Eckart frame and launches the Hessian calculation (carried out by the ab initio code). The file *process.txt* logs the *fdacc* execution.
3. The third execution will retrieve useful information about the vibrational analysis. Harmonic normal mode frequencies are printed in the *fdacc.out* file. At this stage, the code computes the Coriolis contributions. Once this step is completed, it is important to check the *fdacc.out* file.
4. The fourth execution launches the anharmonic constants calculations. Specifically, given N-atom molecule with $N_{\text{vib}} = 3*N - 6(5)$ a total of $1 + 6*N_{\text{vib}}^2 + 8*N_{\text{vib}}*(N_{\text{vib}} - 1)*(N_{\text{vib}} - 2)/6$ SPE calculations will be launched.
5. When the calculations are completed, the last *fdacc* execution will compute the anharmonic constants and copy the output file *X_tot.dat* in your working folder.

Note: all the subsequent runs of the *fdacc* command will work if you are calling it from the correct working folder. If you have your input file in the folder /DIR, then all the *fdacc* commands must be launched from this folder.

Standard execution with option *fin diff = yes*:

With this option the code computes the Hessian using a centered second order approximation with a standard step size of 0.001.

With this option an extra *fdacc* execution is needed between step 2 and 3 of the standard procedure described above.

Execution with additional input files

Execution with an *iso.dat* file

To execute the anharmonicity matrix calculation with isotopic substitutions, place an *iso.dat* file in the launching folder. The calculation will use the defined masses overwriting the geometry specification given in the *input.dat* file. Be aware not to specify the isotopic substitution in the geometry section of the latter file. The *iso.dat* file has the following format:

```
Atom_index   isotope_number
```

```
Atom_index    isotope_number
...
```

the fdacc examples directory includes a sample iso.dat file.

Starting from an already optimized geometry with the R.out or the TS.out files

To start from a previously optimized geometry you need to provide an output file called TS.out or R.out in the working directory. To get reliable anharmonicity matrix results we suggest to carry out the optimizations with stringent convergence criteria (e.g. use the keyword VeryTight when working with Gaussian16) parameters. Note, it is important to define the geometry in the input.dat file following the same atom order used for the optimized geometry.

Execution with the ORDER.dat file

The program will print the harmonic frequencies in ascending order, with the imaginary frequency first for TS calculations, and the anharmonic constants matrix will be ordered accordingly. The user can change this order by specifying the ORDER.dat file structured as follows:

```
Old_NormalMode_index New_NormalMode_index
Old_NormalMode_index New_NormalMode_index
...
```

Output Reading

The fdacc output files are three: fdacc.out, process.txt and X_tot.dat.

- The file **X_tot.dat** is printed during the last iteration of the program and contains the lower triangular matrix of the anharmonic constants. The format is the one suitable for the paradensum, parsctst, and bdens input files.
- The file **process.txt** is updated during each step of the fdacc procedure. It contains information regarding what the program is doing step by step. Furthermore, if any error occurs, here you will find more information about what happened.
- The file **fdacc.out** is updated with useful information to carry out the SCTST calculations with the parsctst code: the harmonic and anharmonic zero point energies, the G_0 term. Furthermore, it

contains the equilibrium geometry in the Eckart frame and the harmonic frequencies. It is highly recommended to check the `fdacc.out` before launching the most computational demanding anharmonic constant calculations.

Important considerations and recommendations

- In the calculation of the Finite Difference (FD) Hessian we set a default value of 0.001 for the FD Cartesian displacement. This is the smallest value we would recommend. Sometimes, when strange or unexpected frequencies are obtained, it may be useful to increase it. A good practice would be to verify the stability of the harmonic frequency values as the step is gradually increased. To change this default option you must modify the `fdacc/src/HESS/calc.f90` and `fdacc/src/HESS/hess.f90` changing respectively the *del* value on line 9 and 11. After the modification you have to recompile these Fortran programs. Then, enter the HESS folder and run the following command:

make all

- If your calculations have problems you can check the scratch folder that will be created inside the `fdacc` folder.

During the calculations the following error may appear: "Error! Something went wrong in the Gaussian calculation!". Always check in the scratch directory what happened to your calculations and eventually relaunch them. The scratch directory path is: `fdacc/scratch/Job_name` (`Job_name` is specified in the `input.dat` file) The program prints the error anytime the standard error file, `myJob.err`, is not empty. If the message printed in `myJob.err` is harmless, delete it and run the *fdacc* program again.

- If you wish to change the standard step size for the anharmonic constant calculations (the user should check the stability of the final rate constant value with respect to this parameter) you can do it by opening the `prog_core_2` file in the `fdacc` folder at the line 19 and changing the *displace* value.

- If you want to update the script with an additional interface to an ab initio code, you can edit your own functions in the fdacc/Interfaces folder. Future releases are foreseen to include more ab-initio programs interfaces. If you need something specific contact the authors.
- If you need to add or modify masses for specific applications you can update the internal fdacc database (function *a_mass*) in the functions_2.sh file.

References

- (1) Mandelli, G.; Aieta, C.; Ceotto, M. Heavy Atom Tunneling in Organic Reactions at Coupled Cluster Potential Accuracy with a Parallel Implementation of Anharmonic Constant Calculations and Semiclassical Transition State Theory. J. Chem. Theory Comput. 2022, 18, 623-637.