

DEVELOPMENT OF A PACKAGE FOR AUTOMATING THE PIPELINE PROCESSING OF QUANTUM-CHEMICAL CALCULATIONS USING GAMESS ON A CLUSTER WITH A SLURM QUEUING SYSTEM

N. Demidovich, D. Zav'yalov, A. Polikarpova
Volgograd State Technical University, Volgograd, Russia

n.a.demidovich@gmail.com

I. INTRODUCTION

The GAMESS package [1] is designed for computational quantum chemistry using ab initio methods. The package allows you to perform calculations in parallel on both shared and distributed memory systems. The experience of using computational quantum-chemical packages shows that it is often necessary to perform pipeline calculations, that is, such calculations in which a number of sequential calculations are automated and the results of each stage serve as input data for the next. For example, in the simplest case, the geometry of the calculated molecule is first optimized, and then some other calculation in which the coordinates optimized at the first stage serve as the initial ones. Existing wrappers for quantum-chemical packages, such as MacMolPlt, Avogadro and a number of others, only allow visualizing the results of calculations or help to compose input files for calculations. Therefore, it was decided to write our own wrapper for GAMESS, which performs the task of automating and pipelining calculations. At the same time, it is necessary that such a program also be able to work with the cluster queuing system - in our case, Slurm [2].

II. MAIN FORMULAS

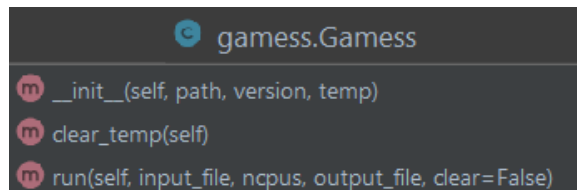
Python was used as the main development language. The following third-party tools were used along with the standard library: ASE (Atomic Simulation Environment) and the Python interface for Open Babel. The first library was used to form the structure of the input file of the GAMESS package; the second is for correct conversion of files into .mol and .gamout formats.

The program is used for calculations of optimized geometry, subsequent energy calculations and report generation. The report is generated in accordance with the requirements of a specific task, therefore, for each task, you can write your own logic for its formation.

The structure of the program is such that pipeline calculations can be done. Before use, you only need to compose a text file with one or several SMILES-structures of chemical compounds and determine the calculation parameters.

During the development, an object-oriented approach was used to provide for further expansion of the program.

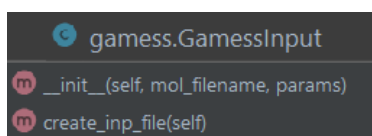
To implement the GAMESS package, the Gamess class was written (Figure 1), which allows you to create an object with initial parameters responsible for the location of the run.gms script, the version of GAMESS and the location of the folder with temporary files. The clear_temp method clears the temporary files folder if necessary; the run method takes the name of the input file for calculations, the number of processors used, the name of the output file and runs the run.gms script using the methods of the standard Python library.



```
class gamess.Gamess:
    def __init__(self, path, version, temp):
    def clear_temp(self):
    def run(self, input_file, ncpus, output_file, clear=False):
```

Figure 1. Gamess class structure

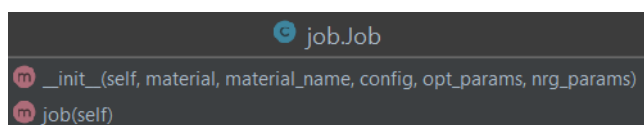
To create an input file for calculations, the GamessInput class was implemented (Figure 2). The object of this class is initialized with parameters that are responsible for the file name in the .mol format and with the calculation parameters.



```
gamess.GamessInput
__init__(self, mol_filename, params)
create_inp_file(self)
```

Figure 2. GamessInput class structure

All the work of the program is performed using the Job class (Figure 3). The object of this class is initialized with the material for which the calculations are carried out, the name of this material (optionally a number), configuration (from a JSON file), parameters of optimization and energy calculations. When calling the job method, the initialization logic of the Gamess class object is executed, the creation of a .mol file, which is converted into an input file for calculations using the GamessInput class object, and the optimization calculations process starts. After that, according to the optimized coordinates, an input file for energy calculations is generated using an object of the GamessInput class, and the process of energy calculations is started.



```
job.Job
__init__(self, material, material_name, config, opt_params, nrg_params)
job(self)
```

Figure 3. Job class structure

The code repository [3] contains a special sbatch.py script that iterates over chemical compounds and runs the job.sh script to perform calculations on a cluster [4] with a Slurm queuing system.

The written package was used for calculations, for example, in works [5,6] and showed that with its help it was possible to significantly save time for organizing calculations and processing intermediate results.

III. CONCLUSIONS

A program has been written to automate chemical calculations using a package for computational quantum chemistry. The program allows you to consistently perform optimization and energy calculations for a group of chemical compounds. To generate reports on the results of calculations and perform calculations on a cluster, the code repository contains special scripts. The development plans of the package include support for other queuing systems and implementation of more pipeline use scenarios.

REFERENCES

- [1] GAMESS — URL: <https://www.msg.chem.iastate.edu/gamess/> (date of visit 09/04/2021).
- [2] Slurm — URL: <https://slurm.schedmd.com> (date of visit 09/04/2021).
- [3] Github-repository — URL: <https://github.com/nidemidovich/gamess-calculations> (date of visit 09/04/2021).
- [4] Cluster used — URL: <https://cluster.vstu.ru> (date of visit 09/04/2021).
- [5] Synthesis and properties of new transparent (co) polyimides based on adamantane-containing diamines and aromatic tetracarboxylic dianhydrides / I.A. Novakov et al. // Russian Chemical Bulletin. - 2021. - Vol. 70, issue 6. - P. 1141-1148. - DOI: <https://doi.org/10.1007/s11172-021-3196-1>.
- [6] Adamantylation of N-aryl and N-arylalkylacetamides in trifluoroacetic acid / I.A. Novakov et al. // Russian Chemical Bulletin. - 2020. - Vol. 69, Issue 6. - P. 1096-1101.