Code and Domain Review Review for Josh Larsen, MODFLOW-API-AG-Package

Michael Fienen, UMid WSC

Code Review

The following items were evaluated in the code review. My findings are in included in *italics* below each item.

1. Coding standards

    a. *I suggest adding flopy and moflowapi to the `environment.yml` file to make it easier for users to get all dependencies installed. I added them in my PR and added a note in the `README.md` file suggesting that*

    b. *The `documentation.md` file has a couple errors. Specifically, in the `Block: PERIOD` explanation of variables, the example string like `<pname1> <id1> <pname2> <id2> <irr_eff> <app_eff>` is missing `iper` and both `irr_eff` and `app_eff` seem to have been renamed in the explanation as `eff_fact` and `app_fact`, respectively. The explanations seem technically correct - just misnamed.*

    c. *There are some issues with the example problems not working. I outline two below, but running others showed similar problems. Rather than outline each one, I just indicate that, in general, I would expect them to work properly. Further, the examples are not documented in the code or in an interpretive file, so it's a bit difficult to know what to make of them.*
    `docs_example.py` is failing, I think, because the tests for `flopy` are not run on installation, so the test files that are relied upon in the `flopy` package to run this example were not available on this fresh installation.
    `mf6_ag_prudic.py` failed to run because of the following

    ```
    File "C:\Users\mnfienen\Miniconda3\envs\modflowapi\Lib\ctypes\__init__.py",
    line 376, in __init__
    self._handle = _dlopen(self._name, mode)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^
    ```

```
FileNotFoundError: Could not find module
'C:\Users\mnfienen\Miniconda3\envs\modflowapi\Lib\site-packages\
mf6api_ag\..\bin\libmf6.dll' (or one of its dependencies).
Try using the full path with constructor syntax.
```

    d. _In general, the docstrings could use a bit more detail and be explicit about whether functions return values or not.

    e. *Similarly, the documentation provides the equations used and a good summary of the techniques, but a few variables do not really include the context to know how a user should choose their values. In particular, these include* **eff_fact**, **app_fact**, *and K_c. I can make assumptions, but references to some more discussion would be helpful.*

2. Unit tests passing *All unit tests passed, although only when running on Windows. The python code and the MODFLOW-API should also run on Linux and Mac, but some of the references were made exploicitly to DLLs for dynamic libraries so that would have to change to run on other operating systems. I recommend either adding explicit support for other operating systems or indicating in the* **README.md** *file that the code is currently only tested on Windows.*

3. User input cleansing
The entry points to this code are through files or other python scripts/notebooks. As a result, there is no outward (e.g. internet) facing interface that could be vulnerable to hacking through input code injection.

4. Memory leaks
As this code is written in python, there are not really ways for memory leaks to be encountered. If the underlying python packages or MODFLOW-6 API exhibit memory leaks, they could be manifest in this code, but that's beyond the scope of this code.

5. Vulnerabilities
No vulnerabilities were identified. The few package dependencies are very popular and thus well-policed by conda and pypi such that malicious code would be unlikely to be introduced through installation of dependencies.

6. Optimizations
I don't see opportunities for clear performance improvement.

Domain Review

1. Comparing output with external data sets
The unit tests include regression tests with a couple existing formulations and I can confirm that the calculations agree.

2. Comparing algorithms with published, scholarly articles about the algorithm
   Many comparisons are made in testing and examples with previous formulations by Dave Prudic. It would be nice to include a reference to those as well.
3. Reviewing unit and integration test results
   As I mentioned above, the tests pass and provide good comparisons with existing formulations.