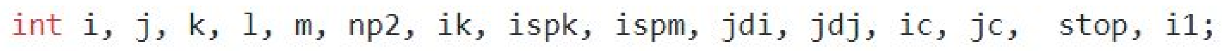
Group 40

Legibility:
Given that the C code was created from fortran there are some issues with the readability. A lot of the variables are like copied from the fortran so they look something like this



```
int i, j, k, l, m, np2, ik, ispk, ispm, jdi, jdj, ic, jc,  stop, i1;
```

There are comments explaining the code at the higher level but it might describe a section like: “Compute element matrices for flow” but following along the code is very difficult. They did a good job of translating the code into C by hand which makes the flow much clearer than its fortran equivalent but still suffers from some things like each file being one long function with lots of variables being declared at the top and then used throughout. Refactoring all of this from fortran could have been completely impossible given the vast amount of code so it does a good job given that.

Implementation:
There is probably a shorter/easier/cleaner way to to write functionally equivalent code. The problem is this would require a deeper understanding of the fortran code which is not something the group could have known given what they were given. The math is also advanced and not something the people in their group understand so they have to copy the functionality of the code more exactly than if they could understand it at a higher level, but it is not their fault. The implementation does a good job with what they have at their disposal, and using C means the code will maintain the speed from fortran.

Maintainability:
The code would be very hard to jump in and maintain given the length and legibility. The code is more maintainable than it was before. They took code from a language that was almost impossible for most people to run and translated it to a widely used language. They also tried to use C features to copy the functionality of larger pieces of code rather than translating statement by statement. Nevertheless, just given the nature of the project, it will be pretty hard to maintain.

Requirements, the code seems to fit the requirements in the requirements document if they can get rid of the remaining bugs. It seems like it is a bit hard to test given they were only given a single test case. But given the complexity of the model one test case passing would almost certainly mean functioning code.

Other: none