

Our program implements the state design pattern to interact with the user. As the user responds to printed prompts, the internal state of the program ("Solver") changes appropriately, whether to save inputted data, print the next prompt for further instructions from the user, or to ask the user to retype something that was incorrectly formatted and not recognized.

Once the user has indicated whether they want to solve a Stokes or Navier-Stokes formulation, the program remains in the same general state, but the "sub-state" ("InputData") changes as the different required pieces of data are entered by the user. Different parser functions are called to determine if the strings typed by the user contain appropriate data in the correct format. All of the data is stored according to the memento pattern. The memento object eventually contains all the data required to solve a formulation. It can be saved to file and reloaded later to plot or make further refinements.

Separate functions can be called to solve, refine, or plot the formulation stored in a memento once it is initially created. These functions are passed the data only once it is all collected by the memento in a list, and thus are never passed incorrectly formatted data since the data format is checked before adding it to the memento.

When creating inflow conditions, the user inputs a string that is then passed to the method `stringToFunction()`. Because of a lack of knowledge about regular expressions at the beginning of working on regular expressions, the `stringToFunction(string function)` method operates in a different way. First the string function is passed to a method `addParen()` that returns a string with parentheses appended around every pair of elements around an operator ($x+y*z$ becomes $(x+(y*z))$). After receiving this string, the method proceeds to step through the string one character at a time using a state machine to call different methods depending on the state. The crux of the conversion into a Function object is pushing functions onto a function stack and pushing operations onto an operation stack as characters are read. Whenever a right parentheses is read, the top two functions and top operation are popped and used to calculate one new function to be pushed back on the function stack. In this manner, the function for the whole string is determined and all exceptions are handled appropriately.